

高等学校信息安全专业规划教材

信息系统安全

陈萍 张涛 赵敏 编著

清华大学出版社

21世纪高等学校信息安全专业规划教材

信息系统安全

陈 萍 张 涛 赵 敏 编著

清华大学出版社
北 京

内 容 简 介

本教材以教育部《信息安全类专业指导性专业规范》所列知识点为基础，从信息系统体系结构层面系统描述信息系统的安全问题及对策。

全书共12章，第1章介绍信息系统安全的基本概念、发展历史、主要目标和技术体系；第2章介绍密码学基本理论与应用，具体包括对称密码体制和公钥密码体制，以及基于密码学的消息认证、数字签名、公钥基础设施PKI；第3章介绍物理安全技术；第4、第5章分别介绍身份认证和访问控制技术；第6章介绍操作系统安全机制；第7章介绍数据库安全技术；第8章介绍信息安全评估标准和我国的信息安全等级保护制度；第9章介绍信息系统安全风险评估的概念、工具、方法、流程；第10章介绍恶意代码的检测与防范技术；第11章介绍软件安全漏洞及防范措施，重点介绍Web应用安全机制；第12章介绍信息安全新技术。

本书可以作为信息安全专业、信息对抗专业、计算机专业、信息工程专业和其他相关专业的本科生和研究生教材，也可作为网络信息安全领域的科技人员与信息系统安全管理员的参考书。

图书在版编目(CIP)数据

信息系统安全/陈萍，张涛，赵敏编著. —北京：清华大学出版社，2016
21世纪高等学校信息安全专业规划教材
ISBN 978-7-302-42227-3

Ⅰ. ①信… Ⅱ. ①陈… ②张… ③赵… Ⅲ. ①信息系统—安全技术 Ⅳ. ①TP309

中国版本图书馆CIP数据核字(2015)第279084号

责任编辑：黄 芝 薛 阳
封面设计：杨 兮
责任校对：梁 毅
责任印制：

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦A座 **邮 编**：100084
社 总 机：010-62770175 **邮 购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：
装 订 者：
经 销：全国新华书店
开 本：185mm×260mm **印 张**：21 **字 数**：523千字
版 次：2016年4月第1版 **印 次**：2016年4月第1次印刷
印 数：1～ 000
定 价： .00元

产品编号：065214-01

前　　言

随着计算机和网络技术的日益普及和广泛应用，信息的应用和共享日益广泛且更为深入，各种信息系统已经成为国家基础设施。与此同时，计算机信息系统的安全问题日益突出，情况也越来越复杂，针对计算机信息系统的攻击与破坏事件层出不穷，如果不对其加以及时和正确的保护，这些攻击与破坏事件轻则干扰人们的日常生活，重则造成巨大的经济损失，甚至威胁到国家安全，所以信息系统的安全问题已经引起许多国家的高度重视，社会对信息安全人才的需求越来越迫切。

确保信息系统安全是一个整体概念，解决某一信息安全问题通常要综合考虑硬件、系统软件、应用软件、管理等多层次的安全问题，目前市面上信息安全方面的书籍大多侧重于网络安全，而专门从信息系统体系结构层面讲解信息安全的教材较少，不利于相关课程教学的实施。

本书以教育部《信息安全类专业指导性专业规范》所列知识点为基础，从信息系统的组成要素出发，寻求综合解决信息安全问题的方案。信息系统自底向上由物理层（硬件层）、操作系统、网络、数据库、应用系统等构成，只有从信息系统硬件和软件的底层出发，确保信息系统各组成部分的安全，从整体上采取措施，才能确保整个信息系统的安全。因此，教材内容以保障信息系统各组成层次安全为一级主线，以各类安全技术在信息系统不同层次上的应用为二级主线进行优化重组，系统地介绍信息系统安全的基本概念、原理、技术、知识体系与应用，覆盖了信息的存储、处理、使用、传输与管理整个生命周期不同环节的安全威胁与相应的保护对策。

本书内容共有 12 章，第 1 章介绍当前信息系统安全形势、信息系统安全的基本概念、发展历史、主要目标和技术体系；第 2 章介绍密码学理论与应用，具体包括密码学的基本概念、密码体制的组成、分类以及设计原则、古典密码体制、对称密码体制、公钥密码体制等密码学基本理论，此外本章还简单介绍了密码学的应用，具体包括用于消息完整性校验的消息认证，用于防止信息抵赖的数字签名技术，以及对公钥进行有效管理和提供通用性安全服务的公钥基础设施技术；第 3 章介绍物理层面增强信息系统安全的方法和技术；第 4 章、第 5 章分别介绍信息系统安全基本技术中的身份认证和访问控制，身份认证是信息系统的第一道安全防线，其目的是确定用户的合法性，阻止非法用户访问系统，访问控制是根据安全策略对用户操作行为进行控制，其目的是为了保证资源受控、合法地使用；第 6 章介绍操作系统安全机制，重点介绍了存储保护、审计、最小特权等安全机制以及介绍主流操作系统 Windows 和 UNIX/Linux 操作系统的安全

机制；第 7 章介绍数据库安全机制，重点介绍数据库访问控制、审计、备份恢复、加密等安全技术，并以实际数据库管理系统 SQL Server 和 Oracle 为例，介绍安全技术在产品中应用情况；第 8 章介绍信息安全评估标准和我国的信息安全等级保护制度；第 9 章介绍信息安全风险评估的概念、工具、方法、流程；本书应用系统层安全从两个角度描述，一是防止应用程序对支持其运行的计算机系统的安全产生破坏，二是防止应用程序自身的安全漏洞被利用，这两部分内容分别体现在第 10、11 章，第 10 章介绍病毒、木马、蠕虫三类恶意代码的检测与防范技术，第 11 章介绍软件安全漏洞及防范措施，重点介绍了 Web 应用安全机制；第 12 章介绍信息安全新技术，包括云安全、物联网安全、移动安全等。

陈萍提出了教材的编写大纲，编写了其中第 1、2、3、4、5、6、8、9、10、11 章，赵敏负责编写第 7、12 章，张涛对教材编写提出了建设性的意见和技术支持。

由于作者自身水平有限，必有许多不足甚至错误之处，恳请读者和专家提出宝贵意见。

笔　者

2015.9

目　　录

第1章　信息系统安全概述

随着计算机和网络技术的日益普及和广泛应用，信息的应用和共享日益广泛且更为深入，人类开始从主要依赖物质和能源的社会步入物质、能源和信息三位一体的社会。各种信息化系统已经成为国家基础设施，支撑着电子政务、电子商务、电子金融、科学研究、网络教育、能源、通信、交通和社会保障等方方面面。

与此同时，计算机信息系统的安全问题日益突出，情况也越来越复杂，针对计算机信息系统的攻击与破坏事件层出不穷，如果不对其加以及时和正确的保护，这些攻击与破坏事件轻则干扰人们的日常生活，重则造成巨大的经济损失，甚至威胁到国家安全。很早就有人提出了“信息战”的概念并将信息武器列为继原子武器、生物武器、化学武器之后的第四大武器，所以信息系统的安全问题已经引起许多国家的高度重视，人们不惜投入大量的人力、物力和财力来提高计算机信息系统的安全性。

本章对计算机信息系统安全问题进行了概述，1.1节介绍目前信息系统面临的主要安全威胁，并指出安全问题的根源；1.2节讲述信息系统安全的基本概念、发展历史、目标；1.3节讲述信息系统安全防护的基本原则；1.4节介绍信息系统安全的技术体系。

1.1　计算机信息系统安全问题

1.1.1　飞速发展的信息化

人类社会经历了农业社会、工业社会，现在已经进入到了信息社会，我国是信息大国，我国的信息化进程起步于20世纪80年代，经过30多年的建设，信息化已经具有一定的规模：信息网络（电信网、互联网和广播电视网）成为支撑经济社会发展的重要基础设施；信息产业成为重要的经济增长点；信息技术在国民经济和社会各领域得到了广泛应用。下面以互联网为例通过一组数据说明我国信息化的发展速度。

据中国互联网信息中心（CNNIC）统计，我国2002年网民数量为5910万人，截至2015年6月份达到了6.68亿，手机网民规模达5.94亿；中国互联网的普及率已经从2005年的8.5%增加到了2015年的48.8%，超过了全球平均水平，中国已经超越美国成为世界上互联网使用人数最多，发展速度最快的国家。

快速发展的信息化引起了人们生产方式、生活方式的巨大变化，极大地推动了人类社会的发展和人类文明的进步，电子政务、电子商务、网络课堂、电子邮件等已经与我们的生活息息相关，利用网络课堂我们可以随时随地聆听世界各地的名师讲课；利用电子商务，我们足不出户就可以购物等。信息化给我们的生活带来了极大的方便，社会对信息系统的依赖性也日益增强。

关于信息化，各国领导人均在不同场合下强调了其重要性：

谁掌握了信息，控制了网络，谁将拥有整个世界——美国著名未来学家阿尔温·托夫勒。

今后的时代，控制世界的国家将不是靠军事，而是信息能力走在前面的国家——美国前总统克林顿。

信息时代的出现，将从根本上改变战争的进行方式——美国前陆军参谋长沙利文上将。

1.1.2 信息安全形势严峻

信息化是一把双刃剑，信息化程度越高，信息安全威胁带来的危害也就越大。据美国金融时报报道，世界上平均每 20s 就发生一次入侵国际互联网的计算机安全事件。据《中国互联网状况》白皮书报道，我国被境外控制的计算机 IP 地址达 100 多万个，被篡改的网站 4.2 万个，被蠕虫病毒感染的计算机每月达 1800 万台，成为世界上被黑客攻击的主要受害国。信息安全已经渗透到国家的政治、经济、军事等领域。2013 年震惊全球的美国中央情报局前雇员斯诺登爆料的“棱镜门”事件，证实了美国一直以来对中国进行着全方位的监控，中国的信息安全形势十分严峻。信息安全对政治、经济、社会稳定、军事等都产生了巨大的影响。

1. 信息安全与政治

目前政府上网已经大规模地发展起来，电子政务工程已经在全国启动。政府网络的安全直接代表了国家形象。从 1999 年到 2001 年，我国一些政府网站曾遭受 4 次大的黑客攻击事件。

第 1 次在 1999 年 1 月左右，美国黑客组织“美国地下军团”联合波兰、英国的黑客组织及其他黑客组织，有组织地对我国的政府网站进行了攻击。

第 2 次是在 1999 年 7 月，李登辉提出两国论的时候。

第 3 次是在 2000 年 5 月，美国轰炸我国驻南联盟大使馆后。

第 4 次是在 2001 年 4 月至 5 月，美国战机撞毁王伟战机并侵入我国南海机场后。

从 2004 年以后，网络威胁呈现多样化，除传统的病毒、垃圾邮件外，危害更大的间谍软件、广告软件、网络钓鱼等纷纷加入互联网安全破坏者的行列，成为威胁计算机安全的帮凶。间谍软件的危害甚至超越传统病毒，成为互联网安全最大的威胁，因此目前，军队以及一些政府机关的计算机是不允许接入互联网的。

2008 年 5 月 1 日，我国颁布施行了《政府信息公开条例》，该《条例》的推行对政府信息化建设提出了新的要求，同时也对电子政务信息系统的网络安全提出了更高的要求。由于政府网站整体安全水平较低，往往是黑客攻击的重要目标，因此，作为政府对外形象的窗口、发布权威信息和与公众开放交流的平台，电子政务信息系统的网络安全管理是一个需要各级部门高度重视的问题。

2. 信息安全与经济

一个国家信息化程度越高，整个国民经济和社会运行对信息资源和信息基础设施的依赖程度也就越高，计算机犯罪造成的经济损失也就越大。1999 年 4 月 26 日，台湾大学生陈英豪编制的 CIH 病毒大爆发，据统计，我国受其影响的 PC 总量达 36 万台之多，经济损失高达 12 亿。2003 年的冲击波病毒（Worm_MSBLAST）造成了全球上百亿的经济损失，2006 年，能将计算机中的所有文件全部变成熊猫烧香图标的熊猫烧香病毒造成了上亿元的经济损失。表 1.1 列举了部分攻击事件造成的经济损失。

表 1.1　攻击事件造成的经济损失

年　　份	攻击行为发起者	损 失 金 额
1998	CIH 病毒	8000 万美元
1999	梅利莎(Melissa)	全球约 3～6 亿美元
2000	Love Letter	88 亿
2001	红色代码	26 亿
2003	Worm_MSBLAST	上百亿
2006	熊猫烧香病毒	上亿
2007	网游大盗	千万

自从 1988 年计算机安全应急响应组(Computer Emergency Response Team,CERT)因 Morris 蠕虫事件成立以来,Internet 安全威胁事件逐年上升,近年来的增长态势变得尤为迅猛,从 1998 年到 2009 年,平均年增长幅度达到了 50%左右,这些安全事件带来了巨大的经济损失,以美国为例,其每年因为安全事件造成的经济损失超过 170 亿美元。

3. 信息安全与社会稳定

在互联网上散布虚假信息、有害信息对社会秩序造成的危害比起现实世界散布谣言所带来的危害更大,严重的会影响社会稳定。

1999 年 4 月,河南商都热线的一个 BBS 上,一张帖子发布虚假消息,说交通银行郑州支行行长携巨款外逃,造成社会动荡,3 天 10 万人上街排队,一天提款 10 多亿元。2010 年 8 月 10 日,一名 17 岁少年在“贴吧”上散布福建泉州地震的谣言,由于发布在知名度很高的网站上,引起全国各地网民关注,造成极为不良的影响,该少年 12 日下午即被警方抓获。

网络上的用户多种多样,尤其是近年来上网的用户越来越多,各种言论和服务很难规范。为了保证网上内容健康,2006 年 4 月 9 日,北京千龙网、新浪网等联合向全国互联网界发出文明办网倡议书,倡议互联网界文明办网,把互联网站建设成为传播先进文化的阵地。

4. 信息安全与军事

信息安全与军事紧密相关,在第二次世界大战中,美国破译了日本人的密码,几乎全歼山本五十六的舰队,重创了日本海军。信息时代的出现,从根本上改变了战争的进行方式,1991 年的海湾战争是一个分界点。1991 年海湾战争前,战争主要以机械化战争为主,而海湾战争后出现了以信息战为主的作战形态。下面介绍几个信息战的重要实例。

海湾战争:是 1991 年 1 月 17 日至 2 月 28 日,以美国为首的多国联盟在联合国安理会的授权下,为恢复科威特领土完整而对伊拉克进行的局部战争。美军通过向带病毒芯片的打印机设备发送指令,致使伊拉克军队系统瘫痪,轻易地摧毁了伊军的防空系统,多国部队运用精湛的信息技术,仅以伤亡百余人的代价取得了歼敌十多万人的成果。海湾战争被称为“世界上首次全面信息战”,充分显示了现代高技术条件下“控制信息权”的关键作用。

科索沃战争:是 1999 年 3 月 24 日至 6 月 10 日,北约对南斯拉夫的空袭行动。在此次战争中美国的电子专家成功侵入了南联盟防空体系的计算机系统,当南联盟军官在计算机屏幕上看到敌机目标的时候,天空上其实什么也没有,通过这种方法,美军成功迷惑了南联盟,使南联盟浪费了大量的人力物力资源。

随着计算机和网络技术的发展，当前的军事战争是信息化战争，信息对抗的攻防能力已成为国防力量之一，利用病毒等作战的典型战例除了前述的海湾战争、科索沃战争外，2007年爱沙尼亚拆除了第二次世界大战苏联将士雕像，引起与俄罗斯的紧张关系，爱沙尼亚遭受网络攻击，政府、报纸银行、企业网站全部瘫痪。2010年为了打击伊朗的核武器制造计划，美国、以色列研发"震网"(Stuxnet)蠕虫，导致伊朗60%的计算机感染，离心机多次出现故障，从而延迟了伊朗的核武器制造计划。

面对严峻的信息安全形势，各个国家纷纷出台各项政策以增强信息系统安全防护能力，减少信息安全问题带来的损失。2003年美国发布了《国家网络安全战略》，正式将网络安全提升至国家安全的战略高度；2009年5月美国奥巴马新政府公布了《网络空间策略评估》，指出美国存在诸多网络安全隐患，表示将制定新的综合方案来保护国家信息基础设施。网络空间又称赛博空间(Cyberspace)，已经成为继陆、海、空、天之后新的战场空间，为了确保在未来网络战中拥有绝对的信息优势，2009年美国成立网络司令部，招募4000名黑客，组建特种部队，该部队主要担负网络攻防任务，并计划于2030年左右完成网络战部队的全面组建，英、俄、日、韩等国也已经组建或正在组建网络战部队。我国是受黑客攻击的主要受害国之一，为了提升网络信息安全防护能力，自2002年起相继召开了"国家信息安全保障体系战略研讨会"、"信息安全保障发展战略研讨会"等，并先后成立了多个研究机构和重点实验室，开展信息安全防护相关技术的研究。

1.1.3 信息系统安全问题的根源

按照我国颁布的《计算机信息系统安全保护等级划分准则》的定义，计算机信息系统是指由计算机及其相关的配套设备、设施(含网络)构成的，按照一定的应用目标和规格对信息进行采集、加工、存储、传输、处理的人机系统。

一个计算机信息系统由硬件、软件以及使用人员三部分组成。其中硬件系统包括组成计算机、网络的硬设备及其他配套设备。软件系统包括操作平台软件、应用平台软件和应用业务软件。操作平台软件通常是指操作系统和语言及其编译系统；应用平台软件通常是指支持应用开发的软件，如数据库管理系统及其开发工具、各种应用编译和调试工具等；应用业务软件是指专为某种应用开发的软件。

信息系统之所以是脆弱的，从技术的角度来看主要原因有以下三个。

1. 网络和通信协议的脆弱性

因特网技术给全球信息共享带来了方便，但是基于TCP/IP协议栈的因特网及其通信协议在设计时，只考虑了互联互通和资源共享问题，存在大量安全漏洞。例如要建立一个完整的TCP连接，必须要在两台通信的计算机之间完成三次握手过程，如图1.1所示，如果三次握手不能够完成，将处于半开连接状态(Half-open)，如图1.2所示，此时服务器端口一直处于打开状态以等待客户端的通信，这个特性往往会被攻击者利用。SYN Flooding拒绝服务攻击就是利用了TCP协议三次握手中的脆弱点进行的攻击，在SYN Flooding攻击中，攻击者向目标机发送大量伪造源地址的TCP SYN报文，这些报文的源地址是虚假的，或者是根本不存在的，当目标机收到这样的请求后，向源地址回复ACK+SYN数据包，由于源地址是假的IP地址，因此没有任何响应，于是目标机继续发送ACK+SYN数据包，并将该半开放连接放入端口的积压队列中，虽然一般系统都有默认的回复次数和超时时间，但由于端

口积压队列的大小有限，如果不断向目标机发送大量伪造 IP 的 SYN 请求，就会形成通常所说的端口被“淹”的情况，使目标机不能提供正常的服务功能。

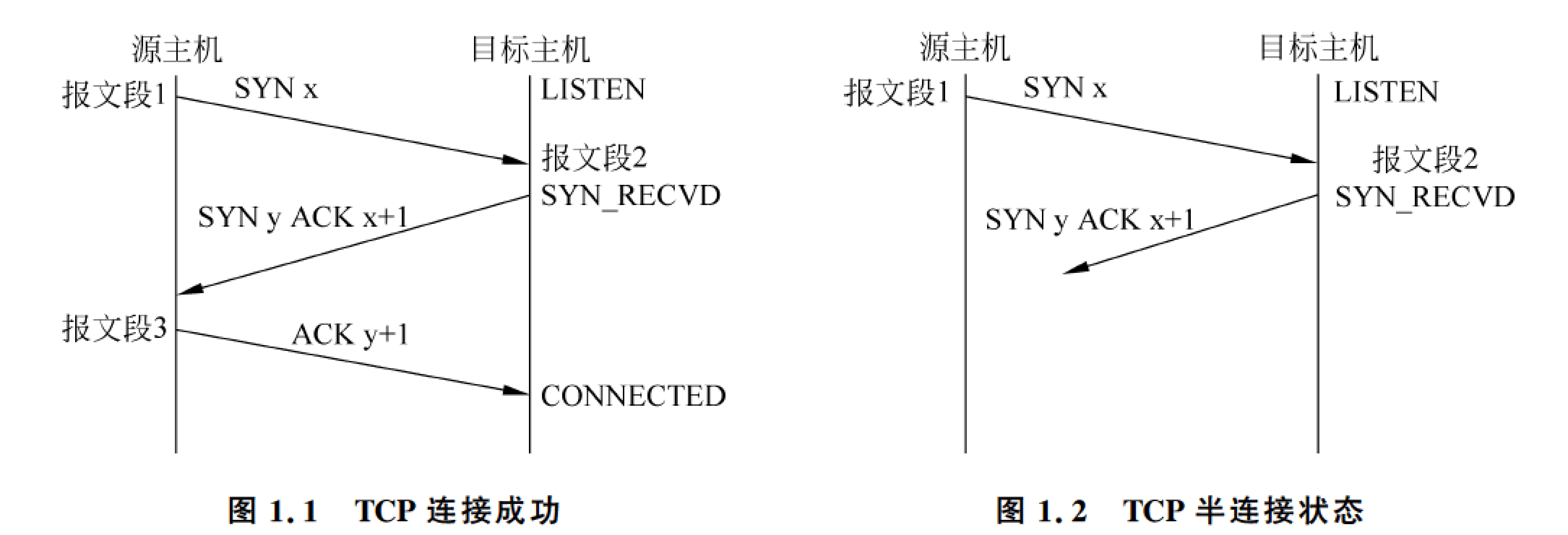


图 1.1　TCP 连接成功　　**图 1.2　TCP 半连接状态**

2. 信息系统的缺陷

信息系统主要由硬件和软件构成，硬软件自身的缺陷客观上导致了计算机系统在安全上的脆弱性。计算机硬件系统由于生产工艺的原因，存在电路短路、断线、接触不良、电压波动的干扰等安全问题；而计算机软件的问题主要受人们认知能力和实践能力的局限性，在系统设计和开发过程中会产生许多错误、缺陷和漏洞，成为安全隐患，而且系统越大、越复杂，这种安全隐患越多。有专家指出：程序每千行中至少有一个缺陷，而目前一个大型软件通常有数百万甚至数千万行语句，这就意味着一个软件可能有几万个差错，随着系统的功能越做越强大，复杂性也不断增加，错误会越来越多。

3. 黑客的恶意攻击

人的因素是影响信息安全问题的最主要因素，人的恶意攻击也称为黑客攻击。早在 20 世纪 60 年代至 70 年代，黑客一词是褒义的，他们是独立思考、奉公守法的计算机迷，典型代表有微软的盖茨、苹果公司的伍茨和乔布斯。当今黑客是指专门闯入计算机系统、网络、电话系统和其他通信系统，具有不同的目的，非法入侵和破坏系统、窃取信息的攻击者。

随着网络的发展，黑客组织越来越扩大化，现在跨地区、跨国界的大型黑客组织已经出现，行动越来越公开化，如每年的黑帽(Black Hat)大会就是全球最大规模的黑客聚会。2008 年 8 月，第 12 届 BlackHatUSA 大会在美国内华达州拉斯维加斯的凯撒皇宫酒店揭开帷幕，在为期两天的会议中，共有来自全球的超过 4500 名参会者针对各种攻击技术进行广泛交流，美国政府也借此会议乘机招募黑客人才。2009 年 3 月，在加拿大温哥华市举行了名为 Pwn2Own 的 2009 年全球黑客大赛，包括 IE8 在内的多套软件被攻破。同时由于存在大量公开的黑客站点，获得黑客工具非常容易，黑客技术也越来越易于掌握，黑客案件越来越频繁化，网络面临的威胁也越来越大。据权威机构调查显示，计算机攻击事件正以年 50%以上的速度增加。如 2008 年上半年，国内外黑客组织对我国网站攻击频繁，平均每天就有数百起网站被黑以及网页被篡改的事件发生，其中来自土耳其的黑客组织 reDMin、sinaritx 等表现活跃。

1.2 信息系统安全的概念

1.2.1 信息系统安全的定义

在讨论信息安全之前，先了解“安全”的含义，安全的基本定义为“远离有危害的状态或特性”或“客观上不存在威胁、主观上不存在恐惧”。中国的《孙子兵法》给出了安全最本质的含义：“用兵之法：无恃其不来，恃吾有以代也；无恃其不攻，恃吾有所不可攻也。”安全问题在各个领域普遍存在，随着计算机网络的迅速发展，人们对信息在存储、处理和传递过程中涉及的安全问题越来越关注，信息领域的安全问题变得非常突出。

传统的信息安全只强调信息本身的安全属性，信息论的基本原理告诉我们，信息不能脱离它的载体而孤立存在，因此不能脱离信息系统而孤立地谈论信息安全，我们应当从信息系统的角度来全面考虑信息安全的内涵。

什么是信息系统安全(the Security of Information System)呢？给信息系统安全下一个确切的定义是比较困难的，因为它包含的内容太过广泛，国际标准化组织(ISO)对信息系统安全提出的建议定义是：“为数据处理系统建立和采取的技术和管理的安全保护。保护计算机硬件、软件、数据不因偶然的或恶意的原因而遭受破坏、更改、泄露”。国内一些学者建议的定义为：“信息系统安全通常是指信息网络的硬件、软件及其系统中的数据受到保护，不受偶然的或者恶意的原因而遭到破坏、更改、泄露，系统连续可靠正常地运行，信息服务不中断”。

信息系统安全主要包括 4 个层面，即设备安全、数据安全、内容安全、行为安全，其中数据安全即传统的信息安全。信息系统设备的安全是信息系统安全的首要问题，包括三个侧面：设备的稳定性(Stability)、设备的可靠性(Reliability)、设备的可用性(Availabity)。数据安全是指采取措施确保数据免受未授权的泄露、篡改和毁坏，这个定义明确了信息系统安全的三个侧面：数据的秘密性(Secrety)、数据的真实性(Authenticity)、数据的完整性(Integrity)。内容安全是信息安全在法律、政治、道德层次上的要求，信息内容在政治上是健康的，必须符合国家法律法规，必须符合中华民族优良的道德规范。行为安全是信息安全的终极目标，确保行为的秘密性、完整性和可控性，行为的秘密性是指行为不能危害数据的秘密性，必要时行为的过程和结果也是秘密的；行为的完整性是指行为不能危害数据的完整性，行为的过程和结果是预期的；行为的可控性是指当行为的过程出现偏离预期时，能够发现、控制或纠正。

信息系统安全必须确保信息在获取、存储、传输和处理各个环节的安全，信息系统硬件安全和操作系统安全是信息系统安全的基础，确保信息系统安全是一个系统工程，只有从信息系统的硬件和软件的底层出发，从整体上采取措施，才能比较有效地确保信息系统的安全。

为了表述简单，在不产生歧义的情况下可以直接将信息系统安全简称为信息安全。

1.2.2 信息系统安全的目标

一个计算机信息系统达到怎样的目标才算安全？为了回答这个问题，我们首先要对信

息系统面临的攻击进行分析。为了获取有用的信息或者达到某种目的，攻击者会采取各种攻击方法对信息系统进行攻击，虽然攻击的表现形式多样，但是从本质上说，这些攻击主要分为两类，即被动攻击和主动攻击。

1. 被动攻击

被动攻击是指攻击者在未被授权的情况下，对传输的信息进行窃听和监测以非法获取信息或数据文件，但不对数据信息做任何修改，通常包括监听未受保护的通信、流量分析、解密弱加密的数据流、获得认证信息等。常用的被动攻击的手段主要包括搭线监听、无线截获、其他截获、流量分析等。

(1) 搭线监听：将导线搭到无人值守的网络传输线路上进行监听。只要所搭的监听设备不影响网络负载，通常不易被发觉，通过解调和正确的协议分析，完全可以掌握通信的全部内容。

(2) 无线截获：通过高灵敏接收装置接收网络站点或网络连接设备辐射的电磁波，通过对电磁信号的分析获得网络数据。

(3) 其他截获：在通信设备或主机中种植木马或病毒程序后，这些程序会将有用的信息发送出来。

(4) 流量分析(Traffic Analysis)：如果由于通过某种手段使得攻击者从截获的信息中无法得到消息的真实内容，攻击者还可以利用统计分析方法对诸如通信双方的通信频度、消息格式、通信的信息流向、通信总量的变化等参数进行监测研究，从中发现有价值的信息和规律。

被动攻击由于不涉及对数据的更改，很难察觉，对于被动攻击的重点在于预防，而不是检测，预防的手段包括加密通信数据等。

2. 主动攻击

主动攻击是指对数据进行篡改和伪造。主动攻击的手段主要分为4类，即伪装、重放、篡改、拒绝服务，如图1.3所示。

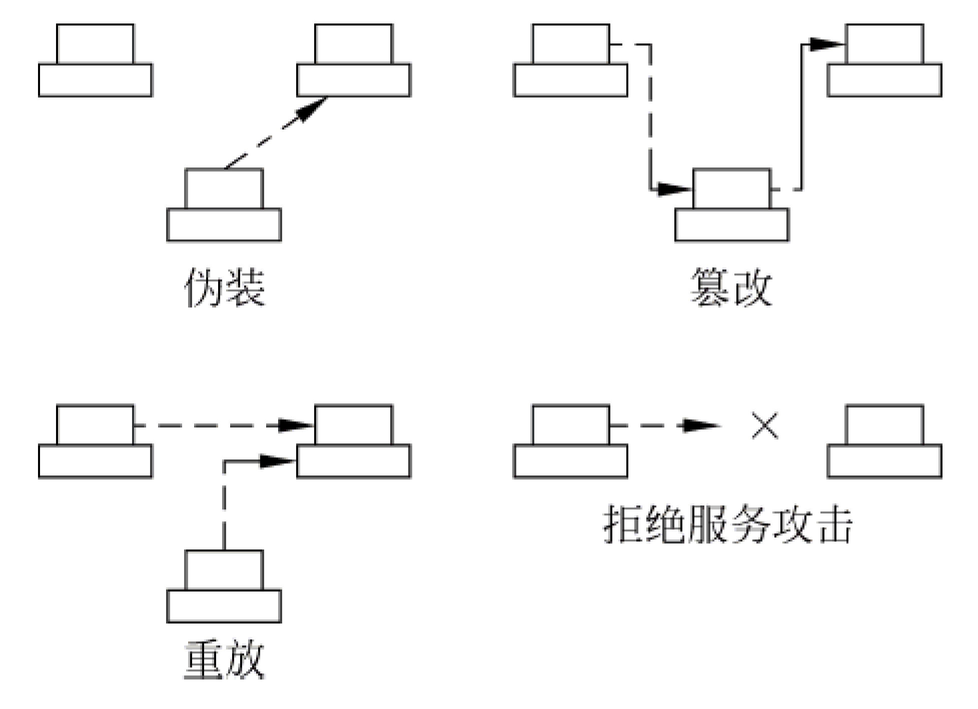


图1.3　主动攻击的4种形式

(1) 伪装：是指一个实体假冒另一个实体，通常攻击者通过欺骗系统冒充成为合法用户以获取合法用户的权限或特权小的攻击者冒充成为特权大的用户。

(2) 重放：攻击者对截获的数据进行复制，并在非授权的情况下进行传输。重放攻击也会带来严重的危害，例如司令员向前方战士发出指令要求前进10m，该指令被截获并被复制，在非授权的情况下，攻击者再次发送这样的指令，使得战士向前行进了20米，由于重放攻击，使得战士多向前行进了10m，这在战争中会带来非常严重的后果。

(3) 篡改：对合法消息的某些部分进行修改、删除，或者延迟消息的传输、改变消息的传输顺序，以产生混淆是非的效果。

(4) 拒绝服务(Denial-of-Service，DoS)：阻止或者禁止信息系统的正常使用，它的主要

形式是破坏某实体网络或信息系统,使得被攻击目标资源耗尽或降低其性能。早期的 DoS 攻击是一对一的攻击,攻击机向目标机发送大量无用数据包,当目标机的 CPU、内存等性能指标比较低时,目标机的资源就会耗费在处理这些无用数据上,而正常的访问请求就会长时间得不到响应。随着计算机处理能力的增长,采用一对一的攻击方式已经不能达到明显的攻击效果,这时分布式拒绝服务(Distribute Denial-of-Service)攻击就应运而生了,分布式拒绝服务攻击采用多对一的攻击方式,采用 10 台或 100 台攻击机同时攻击目标机,以比以前更大的规模来进攻受害者。

主动攻击的特点与被动攻击正好相反,被动攻击虽然难以检测,但是可采取措施有效防止;而要防止主动攻击是十分困难的,因为需要保护的范围太大了,对付主动攻击的重点在于检测并从攻击造成的破坏中及时恢复。

针对主动攻击和被动攻击造成的破坏,提出了信息系统安全的目标。信息系统安全的目标是保护信息的机密性、完整性、可用性、认证和不可否认性,其中机密性是针对信息窃取提出的安全目标;完整性是针对主动攻击中的篡改和重放攻击提出的安全目标;针对伪装攻击,提出了认证的安全目标;针对拒绝服务攻击,提出了可用性的安全目标。上述主动攻击和被动攻击通常是由第三方实施的攻击行为,在信息通信尤其在电子商务中还存在通信的一方由于利益原因抵赖参与过通信的过程,这种行为称为信息抵赖,针对信息抵赖,提出了不可否认性的安全目标,如图 1.4 所示。确保信息系统安全就是要实现上述 5 个目标,其中前三个目标是信息系统安全需要满足的最基本安全目标,简记为 CIA(Confidentiality, Integrity, Availability)。

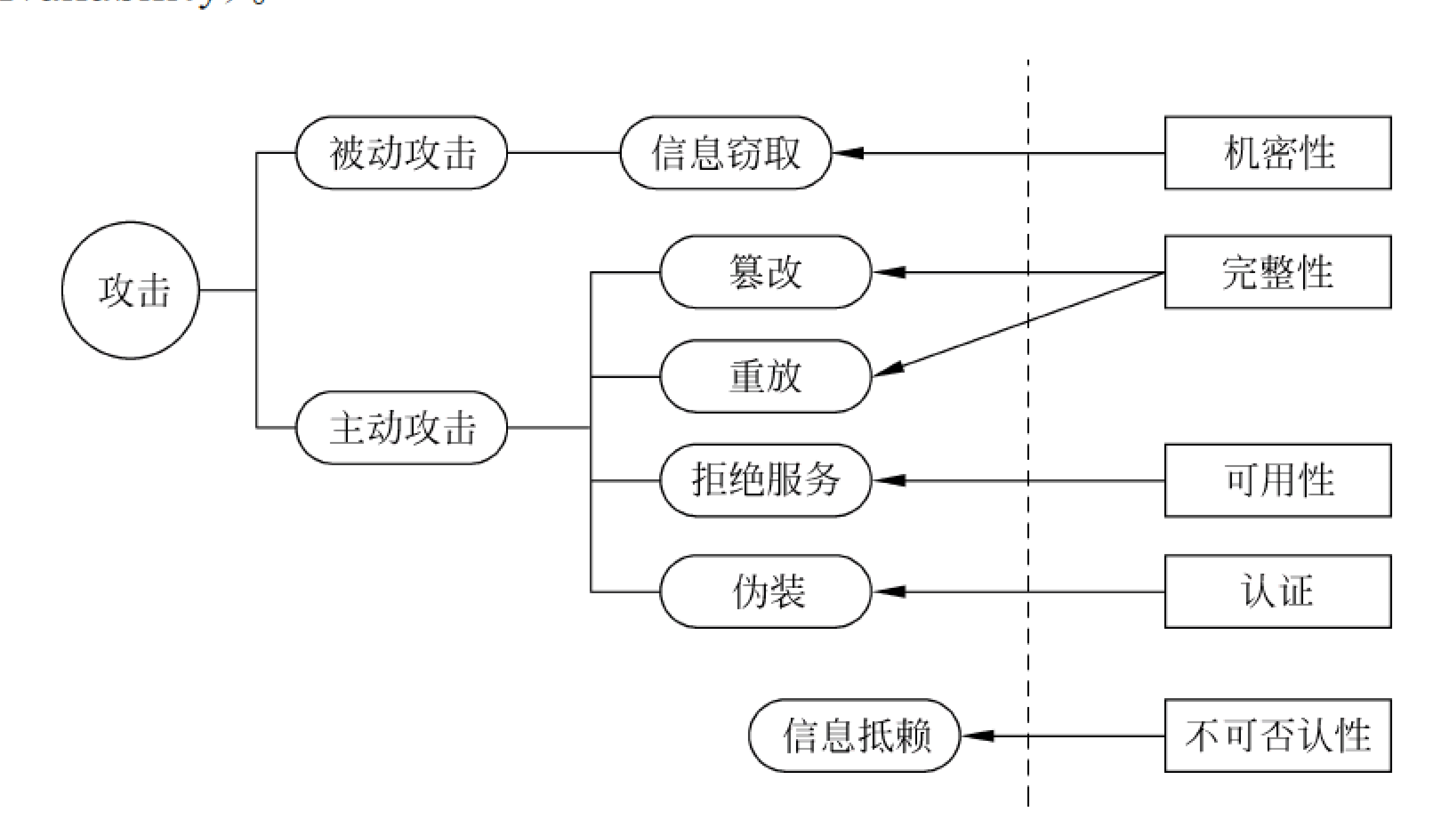


图 1.4 攻击类型与安全目标的关系

1) 机密性(Confidentiality)

机密性是指确保信息不被非授权访问,即使非授权用户得到信息也无法知晓信息内容,有时也称为保密性。

实现机密性的方法一般是对信息加密,或是对信息划分密级并为访问者分配访问权限,系统根据用户的身份权限控制对不同密级信息的访问。针对流量分析攻击导致的信息泄露可通过业务流填充来应对,业务流填充是指在业务闲时发送无用的随机数据,增加攻击者通过通信流量获得信息的困难,它是一种制造假的通信、产生欺骗性数据单元或在数据单元中

填充假数据的安全机制,发送的随机数据应具有良好的模拟性能,能够以假乱真。例如跟平时相比,在发生重大军事行动时,指挥所和作战部队之间的通信量会增加,敌方可以根据通信流量的变化推测某些军事行动的发生,因此为了防止敌方的流量分析,在平时也要发送一些无用的信息。

2) 完整性(Integrity)

完整性是保证信息的真实性,即信息在生成、传输、存储和使用过程中不应发生非授权的篡改、丢失。对于军用信息来说,完整性破坏可能意味着延误战机、闲置战斗力等。实现完整性的方法一般是通过访问控制阻止篡改行为,同时通过消息摘要算法来检验信息是否被篡改。

3) 可用性(Availability)

可用性用来保障信息资源随时可提供服务的能力特性,即授权用户可以根据需要随时访问所需信息。可用性是信息资源服务功能和性能可靠性的度量,涉及物理、网络、系统、数据、应用和用户等多方面的因素,是对信息网络总体可靠性的要求。为了实现可用性,可以采取备份和灾难恢复、应急响应、系统容灾等安全措施。

4) 不可否认性(Non-Repudiation)

不可否认性又称为不可抵赖性,是指信息的发送者无法否认已发出的信息或信息的部分内容,信息的接收者无法否认已经接收的信息或信息的部分内容。实现不可否认性的措施主要有数字签名、可信第三方认证技术等。

5) 认证(Authentication)

认证的目的是通过对用户身份进行鉴别,确保一个实体没有试图冒充别的实体,具体认证手段有口令、智能卡、指纹或视网膜、一次性口令技术、认证协议、多因素认证等。

除此之外,不同的信息系统根据业务类型的不同,可能还有更加细化的具体要求,包括可控性(Controllability)、可审查性(Auditability)、可存活性(Survivability)等。可控性就是对信息及信息系统实施实时监控,确保系统状态可被授权方控制;可审查性是指使用审计等安全机制,使得使用者(包括合法用户、攻击者、破坏者、抵赖者等)的行为有证可查,并能够对网络出现的安全问题提供调查依据和手段;可存活性是指计算机系统的这样一种能力:它能在面对各种攻击和错误的情况下继续提供核心的服务,而且能够及时地恢复全部的服务,这是一个新的融合计算机安全和业务风险管理的课题,它的焦点不仅是对抗计算机入侵者,还要保证在各种网络攻击的情况下业务目标得以实现、关键业务功能得以保持。信息系统安全的目标就是确保这些特性不被破坏。

1.2.3　信息安全的发展历史

信息安全的概念与技术随着计算机、通信与网络等信息技术的发展而不断演化、动态发展,人类对信息安全的认识和观念上的发展经历了通信保密阶段、信息安全阶段和现在的信息安全保障(Information Assurance,IA)阶段。早期的"通信保密"阶段以通信内容保密为主,中期的"信息安全"阶段以信息自身的静态防护为主,而近期的"信息保障"阶段则强调动态的、纵深的、生命周期的、全信息系统资产的信息安全。

1. 通信保密阶段

从古代至20世纪60年代中期，人们更关心信息在传输中的安全，一旦信息在传输过程中被截获，则信息的内容会被敌人知晓。面对通信过程中存在的安全问题，人们强调的主要是信息的保密性，对安全理论和技术的研究也侧重于密码学，这一阶段的信息安全可以简单地称为通信安全，即COMSEC(Communication Security)。

最初，信息传递一般由可靠的使者完成，为了保护传输中的信息，出现了一些朴素的信息伪装方法。北宋曾公亮和丁度合著的《武经总要》反映了北宋军队对军令的伪装方法：先约定40条常用军令，然后用一首含有40个不同字的五言律诗，令其中每个字对应一条军令，传送军令时，写一封普通的书信或文件，在其中的关键字旁加印记，将军们在收到信后，找出其中加印记的关键字，然后根据约定的40字诗查出该字对应的军令。在古代欧洲，代换密码和隐写术得到了较多的研究和使用。德国学者Trithemius于1518年出版的《多表加密》反映了当时欧洲在代换密码的研究上已经从单表、单字符代换发展到了多表、多字符代换。

自19世纪40年代发明电报后，安全通信主要面向保护电文的机密性，密码技术成为获得机密性的核心技术。在两次世界大战中，各发达国家均研制了自己的密码算法和密码机，如在第二次世界大战中，德国使用了一台名为Enigma的机器对发送到军事部门的消息进行加密，此外还有日本的PURPLE密码机与美国的ECM密码机，但当时的密码技术本身并未摆脱主要依靠经验的设计方法，并且由于在技术上没有安全的密钥分发方法，在两次世界大战中有大量的密码通信被破解，以上密码被普遍称为古典密码。1949年，Shannon发表论文《保密系统的信息理论》，用信息论阐述保密通信的原则，标志着密码成为了一门科学。

2. 信息安全阶段

计算机的出现是20世纪的重大事件，深深改变了人类处理和使用信息的方法。20世纪60年代后，半导体和集成电路技术的飞速发展进一步推动了计算机硬件的发展，计算机和网络技术的应用进入了实用化和规模化阶段。此时，对计算机安全的威胁扩展到恶意代码、非法访问、脆弱口令和黑客等，人们对安全的关注已经逐渐扩展为确保计算机系统中的硬件、软件及正在处理、存储和传输信息的机密性、完整性、可用性、认证、不可否认性为目标的信息安全阶段，即INFOSEC(Information Security)。

在密码学方面，美国斯坦福大学的Differ和Hellman于1976年发表了论文《密码学的新方向》，指出不仅密码算法可以公开，加密用的密钥也可以公开，并且公开这些信息不会影响密码系统的安全性，为公钥密码机制提供了理论基础；美国国家标准与技术研究所于1977年通过公开征集的方法制定了当时应用急需的“数据加密标准(Data Encryption Standard，DES)”，推动了分组密码的发展。这两个事件标志着现代密码学的诞生。1978年，麻省理工学院的三位科学家Rivest、Shamir、Adleman设计了著名的RSA公钥密码算法，使数字签名和基于公钥的认证成为可能。20世纪80年代后，学术界提出了很多信息安全新观点和新方法，如椭圆曲线密码(Ellipse Curve Cryptography，ECC)、密钥托管和盲签名等，标准化组织和产业界也制定了大量算法标准和实用协议，如数字签名标准DSS(Digital Signature Standard)、因特网安全协议IPsec(Internet Protocol Security)、安全套接

字层 SSL(Secure Socket Layer),此外形式化分析、零知识证明等都取得了进展。世界各国相继推出了一系列安全评估准则,具有代表性的成果是美国的可信计算机系统评估准则 TCSEC 以及加拿大、法国、德国、荷兰、英国、美国国家安全局于 20 世纪 90 年代中期提出的信息技术安全性评估通用准则(Common Criteria,CC)。

3. 信息安全保障阶段

20 世纪 80 年代末 90 年代初,信息安全领域发生了巨大变化。1988 年美国发生了莫里斯病毒事件,1989 年联邦德国破获了克格勃利用黑客窃取计算机网上秘密案。为此,1989 年美国和西欧国家提出了动态防护的概念,并率先建立计算机应急反应小组。随后,欧美各国建立大量应急组织,并成立"计算机安全应急国际论坛"。1991 年海湾战争后,在 80 年代美、苏、中军事理论界提出信息战命题的基础上,美国正式启动信息战和网络战的研究与准备,引发了世界范围的信息战军备竞赛。与此同时,社会各个重要领域均向信息化迈进,由此带来的针对计算机信息系统的攻击事件日趋频繁,信息安全的概念已经不再局限于对信息的保护。为了保证关键信息系统的安全,系统的安全性和系统的可靠性作为安全的重要内涵引起人们的高度重视。

国际标准化组织(ISO)于 1989 年对 OSI 开放系统互连环境的安全性进行了深入研究,在此基础上提出了 OSI 安全体系结构,1989 年,该标准被我国采用,ISO 7498-2 安全体系结构由 5 类安全服务(认证、访问控制、数据保密性、数据完整性和抗抵抗性)及用来支持安全服务的 8 种安全机制(加密机制、数字签名、访问控制机制、数据完整性机制、认证交换、业务流填充、路由控制和公证)构成。

ISO 7498-2 体系关注的是静态的防护技术,它并没有考虑到信息安全动态性和生命周期的发展特点,缺乏检测、响应和恢复这些重要的环节,因而无法满足更复杂更全面的信息保障的要求。

当各国开始大力发展和建设社会信息基础设施,第一个进入信息化社会的美国,在 1996 年 12 月 9 日以国防部的名义发表了 *DoD Directive S-3600. 1: Information Operation*,在这个命令中,正式提出了信息安全保障(Information Assurance,IA)的概念:"通过确保信息和信息系统的可用性、完整性、可验证性、保密性和不可否认性来保护信息系统的信息作战行动,包括综合利用保护、检测和响应能力以及恢复系统的功能"。

1998 年 1 月 30 日,美国国防部批准发布了《国防部信息保障纲要》,指出信息保障工作应该是持续不断的,它贯穿于平时、危机、冲突及战争期间的全时域,信息保障不仅能满足和平时期的国家信息安全需求,而且能支持战争时期的国防信息安全攻防。同年 10 月,美国国家安全局(NSA)发布了《信息保障技术框架》(Information Assurance Technical Frame,IATF),提出了信息基础设施的整套安全技术保障框架,定义了对一个系统进行信息保障的过程以及该系统中软、硬件的安全需求。IATF 从整体、过程的角度看待信息安全问题,其代表理论是"纵深防护战略"(Defense-in-Depth),依赖人员、技术、操作三个因素最终实现信息保障目标。

(1) 人(People):人是信息系统的拥有者、管理者和使用者,是信息保障体系的核心,是第一位的要素,因此对人的管理在信息安全保障体系中显得尤为重要,安全管理包括安全意识培养、组织管理、技术管理、操作管理等多个方面。

(2) 技术(Technology):技术是实现信息保障的具体措施和手段,这里的技术已经不

单是以防护为主的静态技术，而是保护(Protect)、检测(Detect)、响应(React)和恢复(Restore)有机结合的动态技术体系，也称为PDRR(或PDR²)保障体系，如图1.5所示。

① 保护(Protect)：指采用可能采取的手段保障信息的保密性、完整性、可用性、可控性和不可否认性。

② 检测(Detect)：指提供工具检查系统可能存在的黑客攻击、白领犯罪和病毒泛滥等脆弱性。

③ 响应(React)：指对危及安全的事件、行为、过程及时做出响应处理，杜绝危害的进一步蔓延扩大，力求系统能提供正常服务。

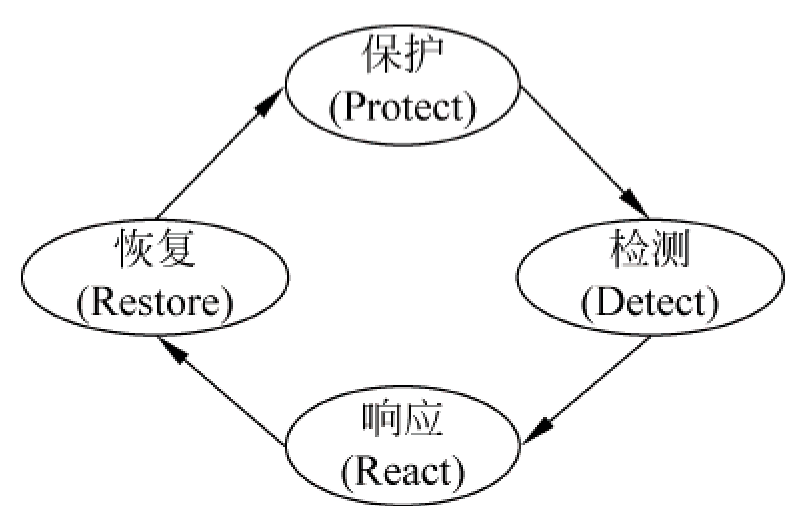


图1.5 PDRR模型

④ 恢复(Restore)：指一旦系统遭到破坏，尽快恢复系统功能，尽早提供正常的服务。

PDRR模型把信息的安全保护作为基础，用检测手段来发现安全漏洞，同时采用应急响应措施对付各种入侵，在系统被入侵后，要采取相应的措施将系统恢复到正常状态，该模型强调自动故障恢复能力。

(3) 操作(Operation)：或者叫运行，操作将人和技术紧密结合在一起，涉及风险评估、安全监控、安全审计、入侵检测、响应恢复等内容。

信息安全保障把信息系统安全从技术扩展到管理、从静态扩展到动态，与前几阶段的信息安全概念和技术相比，层次更高，涉及面更广，提供的安全保障更全面。

在我国，国家信息化领导小组于2003年出台了"国家信息化领导小组关于加强信息安全保障工作的意见"，是我国信息安全领域的指导性和纲领性文件。

1.3 信息系统安全防护基本原则

计算机信息系统面临的安全威胁多种多样，安全威胁和安全事件的原因非常复杂。而且，随着技术的进步以及应用的普及，新的安全威胁不断产生。尽管没有一种完美的、一劳永逸的安全保护方法，但是，如果在设计之初就遵从一些合理的原则，那么相应信息系统的安全性就更加有保障。以下一些安全防护基本原则经过长时间的检验并得到了广泛认同，可以视为保证信息系统安全的一般性方法(或称为原则)。

1. 整体性原则

整体性原则是指从整体上构思和设计信息系统的安全框架，合理选择和布局信息安全的技术组件，使它们之间相互关联、相互补充，达到信息系统整体安全的目标。这就好比用木桶装水，一只木桶装水的容量不是取决于最长的木板而是取决于最短的木板，不仅取决于木板的长度，还取决于木板之间的结合是否紧密，以及这个木桶是否有坚实的底板，这就是著名的"木桶"理论，计算机信息系统安全的研究应该符合这一富含哲理的"木桶"理论。

首先，对于一个庞大而复杂的信息系统，攻击者必然从系统中最薄弱的地方进行攻击。因此，充分、全面、完整地对系统的安全漏洞和安全威胁进行分析、评估和检测(包括模拟攻击)，是设计安全系统的必要前提条件。安全机制和安全服务设计的首要目的是防止最常用的攻击手段，其根本目标是提高整个系统的"安全最低点"的安全性能。

其次,信息安全应该建立在坚实的安全理论、方法和技术的基础之上,这是信息安全的底,通过深入分析信息系统的构成、分析信息安全的本质和关键要素,信息安全的底是密码技术、访问控制技术、安全操作系统、网络安全协议等,它们构成了信息安全的基础。需要花大力气研究信息安全的这些基础、核心和关键技术,并在设计一个信息安全系统时,按照安全策略目标设计和选择这些底部组件,使需要保护的信息安全系统建立在可靠、牢固的安全基础之上。

木桶能否有效地容水,除了需要坚实的底板、相同高度的侧板,还取决于木板之间的缝隙,对于一个安全防护体系而言,安全产品之间的不协同工作犹如木板之间的缝隙,将使木桶无法容水。不同产品之间的有效协作和联动犹如木板之间的桶箍,能把一堆独立的木条联合起来,紧紧地围成一圈,消除木条之间的缝隙,使木条之间形成协作关系,达成一个共同的目标。

2. 分层性原则

没有一种安全技术牢不可破,只要给予攻击者足够的时间和资源,任何安全措施都可能被破解,因此,保障信息系统安全不能依赖单一的保护机制。这就好比确保保存在银行保险箱中的财物安全需要多层安全措施,例如保险箱自身有钥匙和锁具,保险箱置于保险库中,而保险库的位置处于普通人难以到达的银行建筑的中心位置或地下,仅有通过授权的人才能进入保险库,通向保险库的道路有限且又有监控系统,银行大厅有警卫巡视且有联网报警系统。通过不同层次和级别的安全措施共同保证了所存财物的安全。同样在信息系统中只有构建良好分层的安全措施才能够保证信息的安全。

在如图 1.6 所示的信息安全分层保护中,如果一个外部入侵者意图获取最内层主机上存储的信息,必须首先想方设法绕过外部的网络防火墙,突破网络入侵检测系统的识别和检测,登录组织内部网络,此时,入侵者面对的是组织内部的实施网络访问控制的内部防火墙,

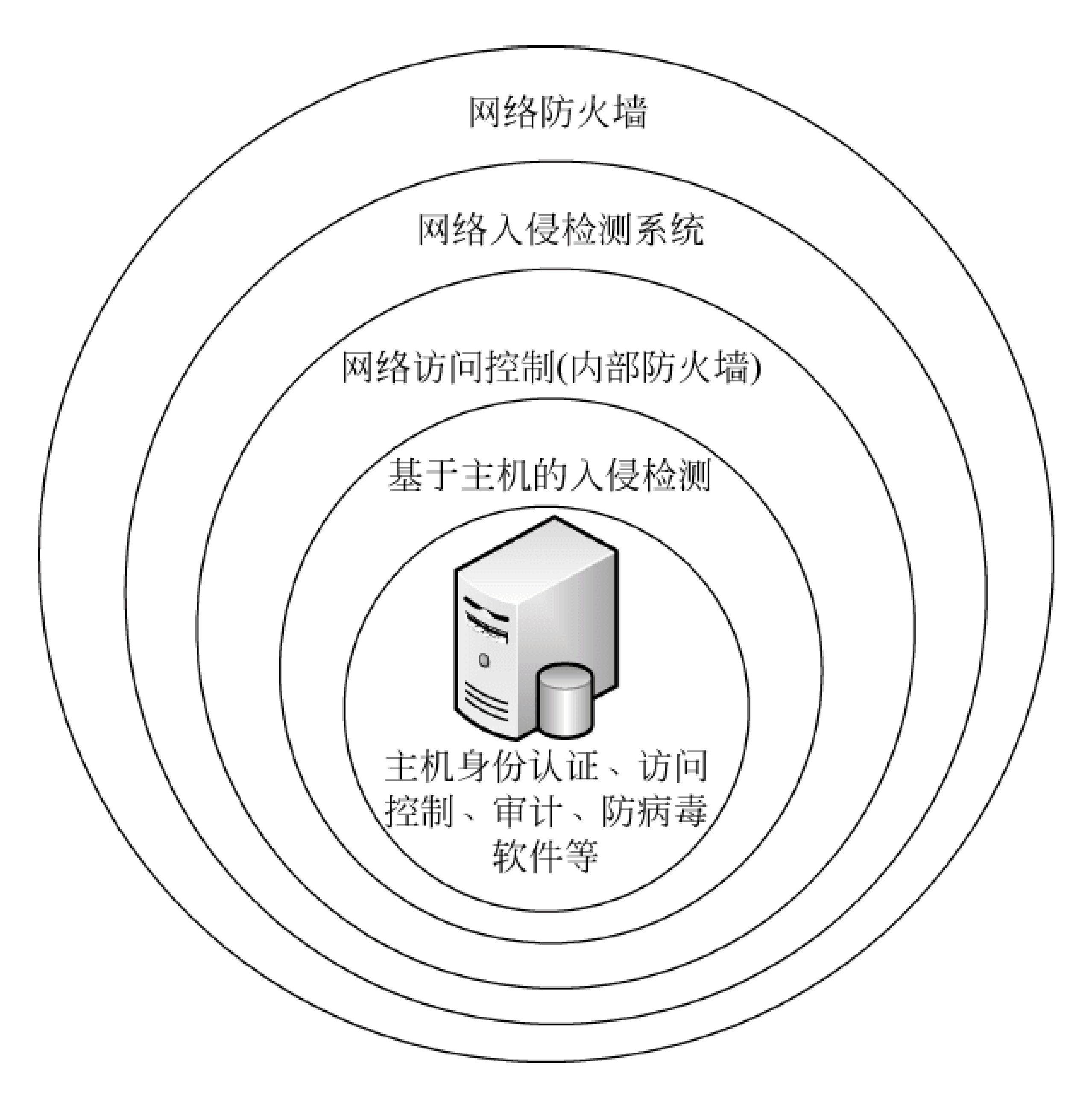


图 1.6　信息系统的分层保护措施

只有在攻破内部防火墙或采用各种方法提升权限后才能进行下一步的入侵，在登录主机后，入侵者将面对基于主机的入侵检测系统，而他也必须想办法躲过检测，最后，如果主机经过良好配置，通常对存储的数据具有强制性的访问控制和权限控制，同时对用户的访问行为进行记录并生成日志文件供系统管理员进行审计，那么入侵者必须将这些控制措施一一突破才能够顺利达到预先设定的目标。即使入侵者突破了某一层，管理员和安全人员仍有可能在下一层安全措施上拦截入侵者。

不同防护层次也会使整个安全系统存在防护冗余，这样即使某一层安全措施出现单点失效，也不会对安全性产生严重影响。因而，提高安全层次的方法不仅包括增强安全层次的数量，也包括在单一安全层次上采取多种不同的安全技术，协同进行安全防范。

在使用分层安全时需要注意，不同的层次之间需要协调工作，这样，一层的工作不至于影响其他层次的正常功能。安全人员需要深刻理解组织的安全目标，详细划分每一个安全层次所提供的保护级别和所起到的作用，以及层次之间的协调和兼容。

3. 最小特权原则

在很多系统中都有一个系统超级用户或系统管理员，拥有对系统全部资源的存取和分配权，所以它的安全至关重要，如果不加以限制，有可能由于超级用户的恶意行为、口令泄密、偶然破坏等对系统造成不可估量的损失和破坏。因此有必要对系统超级用户的权限加以限制，实现权限最小化原则。

最小特权（Least Privilege）的思想是系统不应赋予用户超过其执行任务所需特权以外的特权，或者说仅给用户赋予必不可少的特权，最小特权原则一方面赋予主体“必不可少”的特权以保证用户能完成承担的任务或操作，另一方面它仅给用户“必不可少”的特权从而能限制用户所能进行的操作。同时为了保证系统的安全性，不应对某个用户赋予一个以上职责，而一般系统中的超级用户通常肩负系统管理、审计等多项职责，因而需要将超级用户的特权进行细粒度划分，分别授予不同的管理员，并使其只具有完成任务所需的特权，从而减少由于特权用户口令丢失或错误软件、恶意软件、误操作所引起的损失。

1.4 信息系统安全技术体系

无论在单机系统、局域网还是广域网系统中，都存在着自然和人为等诸多因素的脆弱性和潜在威胁，因此计算机信息系统的安全措施应该能全方位地针对各种不同的威胁和脆弱性，这样才能确保信息的保密性、完整性和可用性。总之，一切影响计算机系统安全的因素和保障计算机信息安全的措施都是计算机系统安全技术的研究内容，信息系统安全是一门涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论、信息论等多种学科的综合性学科。

计算机网络环境下的信息系统安全可以划分为 5 个层次，即物理安全、网络安全、操作系统安全、数据库安全、应用系统安全，如图 1.7 所示。最底层的基础安全技术包括密码技术、身份认证技术、访问控制技术，是各层具体安全技术的基础。

密码技术主要包括密码算法和密码学的应用，具体包括对程密码算法、公钥密码体制、数字签名、消息认证、密钥管理等，它们在不同的场合分别用于提供机密性、完整性、不可否

认性等，是构建信息系统安全的基本要素。

身份认证和访问控制是最基本的安全机制，身份认证的主要目的是确定用户的合法性，阻止非法用户访问系统；访问控制对用户提出的资源访问请求加以控制，其目的是为了保证网络资源受控、合法地使用。这两种基本安全机制可以用在信息系统的各组成部分，确保硬件、操作系统、数据库、应用系统的安全。

应用系统安全技术			
安全编程	恶意代码检测与防御	Web 应用安全	
数据库安全技术			
安全性控制	完整性控制	并发控制	恢复控制
操作系统安全技术			
内存保护	用户标志与识别	授权控制	审计技术
网络安全技术			
防火墙技术	漏洞扫描技术	入侵检测技术	防病毒技术
物理安全技术			
环境安全	设备安全	介质安全	
基础安全技术			
密码技术	身份认证技术	访问控制技术	

图 1.7　计算机网络环境下的信息系统层次结构

(1) 物理安全。物理安全(Physical Security)是为了应对自然灾害、设备自身的缺陷、设备的自然损坏、环境干扰、人为的窃取和破坏，而对计算机设备、设施(包括机房建筑、供电、空调等)、环境、人员、系统等采取的安全措施。

(2) 操作系统安全。操作系统是信息系统的核心组成部分，为整个计算机信息系统提供底层(系统级)的安全保障。操作系统的安全机制包括存储保护、用户认证和访问控制技术等。

(3) 计算机网络安全。主要包括网络安全框架、防火墙和入侵检测系统、网络隔离技术、网络安全协议，以及公钥基础设施 PKI/PMI 等内容。网络安全有专门的理论和技术，相对独立，有很多专门介绍网络安全的书籍，本书不对这部分内容作详细介绍。

(4) 数据库系统安全。数据库中存放了大量关键数据，需要加以保护，主要借助于数据库管理系统提供的安全机制来实现保护，具体安全机制包括身份认证、访问控制、审计、加密、备份和恢复等。

(5) 应用系统安全。主要包括病毒、木马、蠕虫等恶意程序攻击的原理及防范措施；应用系统自身因编程不当存在缓冲区漏洞、格式化字符串漏洞等，如何开发安全应用系统的编程方法、如何确保 Web 应用系统安全。

此外，在 PDRR 模型中，响应和恢复是两个重要的环节，因此本书还介绍计算机系统应急响应与灾难恢复的概念和内容。安全风险评估也是加强信息安全保障体系建设和管理的关键环节，本书介绍了安全评估的国内外标准，评估的主要方法、工具、过程。

1.5 小　　结

信息化是一把双刃剑,信息化程度越高,信息安全威胁带来的危害也就越大。由于信息系统自身的脆弱性、网络黑客的存在使得信息安全攻击事件不可避免、层出不穷,信息安全形势严峻,网络空间已经成为继陆、海、空、天之后的新的战场空间。信息系统安全是一门涉及多学科的综合性交叉学科,从广义上说,凡是涉及到信息的机密性、完整性、可用性、真实性和可控性的相关技术和理论都是信息系统安全的研究领域。信息系统安全概念的发展经历了保密通信、计算机安全、网络安全阶段,现已进入信息安全保障阶段。确保信息系统安全需要遵循整体性、分层性、最小特权原则。信息系统安全需要确保信息系统各组成层次的安全,主要内容包括物理安全、操作系统安全、数据库安全、应用系统安全等。

习　　题

一、填空题

1. 信息安全的基本安全目标包括________、________和________。
2. 消息篡改属于________攻击。
3. 信息系统安全包括 4 个层面,分别是________、________、________和________。
4. 信息安全概念的发展经历了________、________和________三个阶段。
5. PDRR 模型各部分含义分别为________、________、________和________。

二、选择题

1. 信息安全的基本属性是(　　)。

   A. 机密性　　B. 可用性　　C. 完整性　　D. 上面三项都是

2. 从攻击方式区分攻击类型,可分为被动攻击和主动攻击。被动攻击难以(　　),然而(　　)这些攻击是可行的;主动攻击难以(　　),然而(　　)这些攻击是可行的。

   A. 预防,检测,预防,检测　　B. 检测,预防,检测,预防
   C. 检测,预防,预防,检测　　D. 上面 3 项都不是

3. 从安全属性对各种网络攻击进行分类,阻断攻击是针对(　　)的攻击。

   A. 机密性　　B. 可用性　　C. 完整性　　D. 不可否认性

4. 从安全属性对各种攻击进行分类,截获攻击是针对(　　)的攻击。

   A. 机密性　　B. 可用性　　C. 完整性　　D. 上面三项都是

5. 拒绝服务攻击的后果是(　　)。

   A. 信息不可用　　B. 系统宕机
   C. 应用程序不可用　　D. 上面三项都是

6. 攻击者用传输数据来冲击网络接口,使服务器过于繁忙以至于不能应答请求的攻击方式是(　　)。

   A. 拒绝服务攻击　　B. 会话劫持

C. 信号包探测程序攻击　　D. 地址欺骗攻击

7. 攻击者截获并记录了从 A 到 B 的数据，然后又从早些时候所截获的数据中提取出信息重新发往 B，称为(　　)。

A. 中间人攻击　　B. 强力攻击　　C. 重放攻击　　D. 字典攻击

8. 定期对系统和数据进行备份，在发生灾难时进行恢复，该机制是为了满足信息安全的(　　)属性。

A. 机密性　　B. 可用性　　C. 完整性　　D. 不可否认性

9. 信息安全的木桶原理是指(　　)。

A. 整体安全水平由安全级别最低的部分决定

B. 整体安全水平由安全级别最高的部分决定

C. 整体安全水平由各组成部分的安全级别平均值决定

D. 上面三项都不对

10. DoS 破坏了信息的(　　)。

A. 机密性　　B. 可用性　　C. 完整性　　D. 不可否认性

三、简答题

1. 计算机信息系统的脆弱性在哪里？

2. 简述主动攻击与被动攻击的特点，并列举主动攻击与被动攻击的方式。

3. 信息系统的安全目标有哪些，如何理解？

4. 信息安全概念发展的主要阶段有哪些，各阶段中主要的安全技术有哪些？

5. 什么是信息系统安全的“木桶原理”，如何理解？

6. 查阅资料，进一步了解 PDR、PPDR、PDRR 以及 PPDRR 模型中组成部分的含义，这些模型的发展说明了什么，写一篇读书报告说明。

7. 我国正逐步形成一个完善的安全保障体系，成立了国家计算机网络应急处理协调中心(CNCERT，http://www.cert.org.cn)、国家计算机病毒应急处理中心(http://www.antiviruschina.org.cn)、国家计算机网络入侵防范中心(http://www.nipc.org.cn)、信息安全国家重点实验室网站(http://www.is.ac.cn)。请访问以上网站，了解最新的信息安全研究动态和研究成果。

第 2 章　密码学基础

随着计算机的广泛应用,大量信息以数字的形式存放在计算机系统中,并通过公共信道传输。计算机系统和公共信道在不设防的情况下是很脆弱的,面临信息窃取、信息篡改、信息重放、信息抵赖等安全问题,解决这些安全问题的基础是现代密码学。

密码学是关于加密和解密变换的一门科学,是信息安全理论与技术的基石,在信息安全领域发挥着中流砥柱的作用。通过对信息进行加密将可读的信息变换成不可理解的乱码,从而起到保护信息的作用;密码技术还能够提供完整性校验,即能检测收到的消息是否来自可信的源点、是否被篡改;基于密码体制的数字签名具有抗抵赖的功能。

本章讨论密码学基础,2.1 节介绍密码学的起源;2.2 节介绍密码学的基本概念,包括密码体制的组成、分类以及设计原则等;2.3 节介绍古典密码体制,包括代换密码和置换密码的经典算法;2.4 节介绍对称密码体制,对称密码主要提供机密性,当前主要包括分组密码和序列密码(也称流密码),本节主要介绍在信息加密技术发展史上具有里程碑意义的对称分组密码 DES 算法;2.5 节以著名的 RSA 算法为例介绍公钥密码体制;2.6 节介绍用于消息完整性校验的消息认证;2.7 节主要介绍用于防止信息抵赖的数字签名技术;2.8 节介绍对公钥进行有效管理以及提供通用性安全服务的公钥基础设施技术。

2.1　密码学的发展历史

密码学的发展历史极为久远,其起源可以追溯到几千年前的埃及、巴比伦和古希腊,或许是由于最早的密码起源于古希腊,密码学的英文单词 cryptology 一词来源于希腊语,crypto 是隐藏或秘密的意思,logo 是单词的意思,graphy 是书写的意思,cryptology 就是"如何秘密地书写单词"。从传统意义上来说,密码学主要研究如何把信息转换成一种隐蔽的方式从而阻止其他人得到它。密码学提供的最基础的服务就是将信息表述为不可读内容,使通信者能够相互发送消息同时避免其他人员读取信息内容,随着密码学的发展,它还提供了身份认证、完整性校验、数字签名等安全服务。

从远古时期到 1949 年,这段时间是科学密码学的前夜时期,这段时间的密码学更像一门艺术,密码专家们通常凭直觉和信念进行密码分析和设计,而不是依靠严格的推理证明,因此这个时期的密码通信还不能称为一门科学,直到 1949 年,香农发表了一篇题为《保密系统的信息理论》的著名论文,在该论文中首次将信息论引入了密码,用数学方法对信息源、密钥、密文等进行了数学描述和定量分析,提出了通用的保密通信模型,将密码置于坚实的数学基础之上,从而把已有数千年历史的密码学推向科学的轨道,标志着密码学作为一门学科的形成。

受历史的限制,20 世纪 70 年代以前,密码学研究基本上是秘密进行的,密码体制及其设计细节都是保密的,主要应用局限于军事、外交、情报、政府等重要部门。直到 1976 年美

国密码学家 W. Diffie 和 M. Hellman 发表论文《密码学的新方向》，该论文提出了一个崭新的思想：不仅加密用的算法可以公开甚至加密用的密钥也可以公开，并且并不会因为公开这些信息而使信息的保密性降低，这就是著名的公钥密码体制的思想。从 1976 年开始直到现在是密码学的蓬勃发展时期，密码研究从秘密走向公开，并逐渐在民用领域得到广泛应用，从而为其注入了强大的生命力，现代密码学发展史上的主要成果如下。

1. DES 算法

1977 年美国国家标准局颁布了数据加密标准 DES，DES 是最早受到广泛应用和具有深远影响的对称分组加密算法，用于非国家保密机关。该算法的加、解密算法完全公开，算法的安全性基于密钥的保密性，其设计充分体现了香农信息保密理论所阐述的密码设计思想，标志着密码设计和分析达到了新的水平。该算法使用了近 20 年，是密码学上的一个创举。

2. RSA 算法

1978 年，美国麻省理工学院三位年轻的数学家 R. L. Rivest、A. Shamir 和 L. Adleman 提出了 RSA 公钥密码体制，它是第一个成熟的、迄今为止理论上最为成功的公钥密码体制。RSA 算法的安全性基于数论中的大整数因子分解的难题，由于该难题至今没有有效的解决算法，这使得该加密机制具有较高的安全性。

3. DSS 算法

数字签名就是数字形式的签名盖章，是证明当事者身份和确保数据真实性的一种重要措施，用于防范通信双方的欺骗。没有数字签名，诸如电子政务、电子金融、电子商务等系统是不能实际使用的。数字签名一般利用公钥密码体制进行，其安全性取决于密码体制的安全程度，1994 年美国公布了数字签名标准 DSS(Digital Signature Standard)，由于美国在科学技术方面的领先地位，DSS 实际上已经成为国际标准。

4. AES 算法

DES 算法在使用近 20 年后，由于密钥太短，抵制不住穷举攻击，其安全性已无法保证。1995 年美国国家标准与技术研究所公开征集用于取代 DES 的高级加密标准 AES (Advanced Encryption Standard)，最终在 2001 年 10 月，正式采纳比利时密码学家 Joan Daemen 和 Vincent Rijmen 提出的 Rijndael 算法作为 AES 算法，该算法是分组长度和密钥长度均可变的多轮迭代型加密算法，集安全性、效率、可实现性及灵活性于一体。2002 年许多国家标准化组织都采纳 AES 作为加密标准。为了和国际接轨，我国也在某些商业领域中使用 AES。

进入 21 世纪，各种新领域的密码学也广泛开展，随着量子计算机研究热潮的兴起，世界各国对量子密码的研究也广泛开展。量子密码具有可证明的安全性，同时还能对窃听行为方便地进行检测，这些优势使得量子密码引起了国际密码学界的高度重视。另外混沌是一种复杂的非线性非平衡动力学过程，由于混沌序列是一种具有良好随机性的非线性序列，有可能构成新的序列密码，因此世界各国的密码学者对混沌密码寄予了很大的期望。还有生物信息技术的发展也推动着生物芯片、生物计算机和基于生物信息特征的生物密码的研究。量子密码、混沌密码和生物密码的出现将把我们带入新的密码学世界。

2.2 密码学基本概念

2.2.1 密码体制的组成

密码学(Cryptology)包括密码编码学(Cryptography)和密码分析学(Cryptanalysis)两个分支。研究各种加密方案的学科称为密码编码学,加密方案称为密码系统或密码体制;研究破译密码的学科称为密码分析学。密码编码学和密码分析学既相互对立又相互依存,密码学就是对这两个分支进行综合分析、系统研究的科学。

加密的基本思想就是对信息进行伪装,使非法接入者无法理解信息的真正含义。这里伪装就是对信息实施一组可逆的数学变换,我们称伪装前的原始信息为明文(Plaintext),伪装后的信息为密文(Ciphertext),伪装的过程为加密(Encryption),去掉伪装恢复消息本来面目的过程称为解密(Decryption),加密和解密的过程要在密钥(Key)的控制下进行,密钥是只被通信双方所掌握的关键信息。因此,一个密码系统由以下 5 部分组成。

(1) 明文空间(M):全体明文的集合。

(2) 密文空间(C):全体密文的集合。

(3) 加密算法(E):一组由 M 到 C 的变换。

(4) 解密算法(D):一组由 C 到 M 的变换。

(5) 密钥空间(K):全体密钥的集合,其中,加密密钥用 K_e 表示,解密密钥用 K_d 表示。

加密就是明文在加密密钥和加密算法的共同作用下生成密文的过程:$C=E(M,K_e)$,解密就是密文在解密密钥和解密算法的作用下恢复成明文的过程:$M=D(C,K_d)$,如图 2.1 所示。

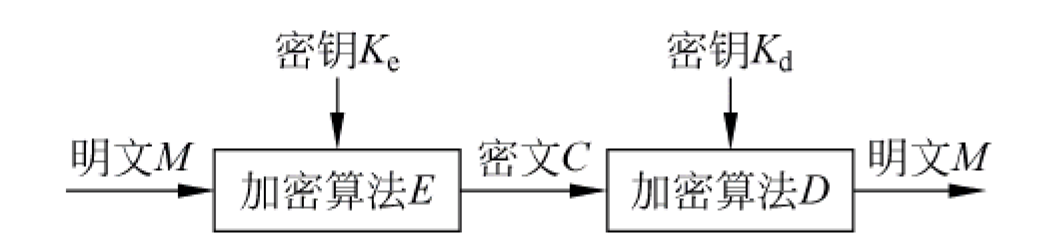


图 2.1 加密和解密过程

荷兰人 Kerckhoffs 在 1883 年指出,密码算法的安全性必须建立在密钥保密的基础上,即使敌手知道算法,若不掌握特定密钥也难以破译密码算法,这就是著名的 Kerckhoffs 准则。因此,数据的安全应该基于密钥的保密而不是算法的保密,也就是说密码体制中的加、解密算法是公开的,可供所有人使用、研究,只有能经受得住敌手充分研究而找不出破绽的算法才是安全的算法,这就是算法设计公开的原则,其目的是使算法设计完备、没有缺陷。美国在制定数据加密标准 DES 算法时就采用公开征集、公开评价的原则,实践证明 DES 算法是安全的。当然,密码设计公开的原则并不要求所有密码在应用时都要公开加、解密算法,例如国家的军政核心密码一般都不公开其加、解密算法,但这些密码在设计时仍然应坚持算法公开的原则,只不过是只对内部专业设计与分析人员公开,而不对外公开。在公开设计原则下是安全的密码体制,在实际使用时对算法保密,将会更安全,这是核心密码系统设计和使用的正确路线。而对于商业密码则应当坚持公开征集、公开评价的原则。

2.2.2　密码体制的分类

1. 根据变换对象分类

根据加密变换对象的不同，密码体制分为古典密码体制和现代密码体制。其中，古典密码体制以字母(字符)作为变换的单位，通常指从古代至第二次世界大战前后产生的密码，目前已经很少使用，而现代密码体制以位或字节作为变换的单位。

2. 根据密钥的使用方式分类

根据密钥使用方式的不同，密码体制可分为对称密码体制和非对称密码体制(也称为公钥密码体制)。

对称密码体制是指用于加密数据的密钥和用于解密数据的密钥相同或者两者之间存在某种明确的数学关系因而容易相互导出。由于算法本身可以公开，因此采用对称加密算法进行加密通信前，需要通过可靠的途径将密钥送至接收端，一旦密钥泄露就等于泄露了被加密的信息。对称密码算法中最广泛使用的是 DES 算法。

非对称加密算法是指用于加密数据的密钥和用于解密数据的密钥是不同的，而且已知加密密钥无法推导出解密密钥。在非对称加密算法中用于加密的密钥是可以公开的，任何人都可以基于公开密钥对信息进行加密，但只有拥有对应解密密钥的人才能解密信息，因此解密密钥需要严格保密。RSA 算法是常用的非对称密码算法。

这两类密码技术都能提供机密性，但对称密码体制的加密效率更高，因此它常用于数据量较大的保密通信中，而公钥密码常用于数字签名、密钥分发等场合。

3. 根据明文和密文的处理方式分类

根据明文处理方式和密钥使用方式的不同，可以将密码体制分成分组密码(Block Cipher)和序列密码(Stream Cipher)。分组密码一次加密一个明文块，序列密码一次加密一个字符或一个位。

分组密码也称为块密码，它将明文 M 划分成一系列明文块 $M_1, M_2, \cdots, M_n$，通常每块包括若干字符，并且对每块 M_i 用同一个密钥 K_e 逐个进行加密，即：

$C=(C_1, C_2, \cdots, C_n)$，其中 $C_i=E(M_i, K_e)$，$i=1,2,\cdots,n$。

序列密码也称为流密码，它将明文和密钥都划分为位(b)或字符的序列，并且对明文序列中的每一位或字符都用密钥序列中对应的分量来加密。即：

$M=(m_1, m_2, \cdots, m_n)$，$K_e=(k_{e1}, k_{e2}, \cdots, k_{en})$，$C=(c_1, c_2, \cdots, c_n)$，其中 $c_i=E(m_i, k_{ei})$，$i=1,2,\cdots,n$。

2.2.3　密码设计的两个重要原则

直到第二次世界大战，密码技术仍较为简陋。为了寻求新的密码设计方法，SHANNON 于 1949 年提出密码系统设计的两个基本原则。

1. 扩散性(Diffusion)

密码系统应该把明文或密钥信息的变化尽可能多地散布到输出的密文信息中，以便隐蔽明文信息的统计特性。产生扩散的最简单的方法是通过置换(例如重新排列字符)。

2. 混淆性(Confusion)

混淆是指密码系统应该在加密变换过程中使明文、密钥及密文之间的关系复杂化,用于掩盖明文和密文间的关系。产生混淆通常采用的方法是代换。

随后出现的对称密码主要包括分组密码和序列密码,它们的设计者用不同的方法贯彻了以上原则。分组密码有很多,如 DES、AES、IDES、RC5、SMS4 等。SMS4 是 2006 年中国国家密码管理局公布的无线局域网产品中应用的建议密码算法之一。

2.2.4 密码分析

密码分析学是在不知道密钥的情况下,恢复出明文的一门科学。成功的密码分析能恢复出消息的明文或密钥,密码分析也可以发现密码体制的弱点,是评判密码系统的安全性的重要方法。在密码学术语中,“分析”与“攻击”意义相近,因此密码分析也称为密码攻击。在所有密码分析中,均假设攻击者知道正在使用的密码体制。密码分析的方法有两类,即穷举法和分析法。

1. 穷举法

穷举法又称为蛮力攻击,是指密码分析者对截获的密文用所有可能的密钥试译,直到找到一个正确的密钥能够把密文还原成明文,这就是穷举攻击。对于穷举攻击,只要有足够的计算时间,原则上总能成功,当然在穷举法中也要用到经验、直觉判断、猜测等能力。平均而言,破译成功至少要尝试所有可能密钥的一半。穷举攻击所花费的时间等于尝试次数乘以一次解密(破解)所需要的时间。显然可以通过增大密钥量或加大解密(加密)算法的复杂性来对抗穷举攻击。当密钥量增大时,尝试的次数必然增大,当解密(加密)算法的复杂性增大时,完成一次解密(加密)所需的时间增大,从而使穷举攻击在实际上不能实现。

2. 分析法

分析法又分为统计分析攻击和数学分析攻击。统计分析是指密码分析者对截获的密文进行统计分析,利用密文的统计规律来破译密码。例如单表代换密码通过分析密文中字母出现的频率,与英文字母的使用频率做对比而进行攻击。对抗统计分析攻击的方法是增加算法的混淆型和扩散性,打破密文的统计规律。数学分析法是指密码分析者针对加密算法的数学依据,通过数学求解的方法来破译密码。为了对抗数学分析攻击,应选用具有坚实数学基础和足够复杂度的加密算法。

根据密码分析者可利用的数据,密码分析可分为以下几种类型。

(1) 唯密文(Ciphertext Only)攻击。是指密码分析者仅根据截获的密文来破译密码。密码分析者有一些消息的密文,这些消息都是用同一加密算法加密。密码分析者的任务是恢复尽可能多的明文,或者最好是能推算出加密消息的密钥,以便可采用相同的密钥解出其他被加密的消息。

(2) 已知明文(Known Plaintext)攻击。密码分析者可得到一些消息的密文,而且也知道这些密文对应的明文。分析者的任务是用已知信息推出用来加密的密钥或导出一个算法,该算法可以对用同一密钥加密的任何消息进行解密。近代密码学认为,一个密码仅当它能经受得住已知明文攻击才是可取的。第二次世界大战中的中途岛海战是一次成功的已知明文的攻击,美军故意透露出假情报(明文)来诱使日军发报(密文),从而得知密文中日本海军下一个攻击目标“AF”指的是中途岛。

(3) 选择明文(Chosen Plaintext)攻击。指密码分析者不仅可以得到一些小的密文和相应的明文,而且他们也可以选择被加密的明文,这是对密码分析者最有利的情况。例如公钥密码体制中,攻击者可以利用公钥加密任意选定的明文,这种攻击就是选择明文攻击情况。计算机文件系统和数据库特别容易受到这种攻击,因为攻击者可以随意选择明文,并得到相应的密文文件和数据库。

(4) 选择密文(Chosen Ciphertext)攻击。密码分析者能选择不同的被加密的密文,并可得到对应的解密后的明文。

在这些攻击方法中,唯密文攻击难度最大,因为攻击者拥有的信息量最少。

如果密码分析者无论具有多少资源,都不足以唯一地确定在该密码体制下的密文所对应的明文,则此加密体制是无条件安全的。一次一密(One Time Pad)可以满足无条件安全,它用一组完全无序的数字(密钥)对消息进行加密,而且每个密钥只使用一次。除了一次一密,几乎所有的算法都不是无条件安全的,也就是说,在理论上是可能被攻破的,在密码研究中更关心的是在计算上安全的密码体制,加密算法应该至少满足下面两个条件之一就认为是计算上安全的:

(1) 破译密码的代价超出密文信息的价值。那么对于破译密码的人来说,这么做是没有意义的。

(2) 破译密码的时间超出了密文信息的有效期。当密码被破解时,明文实际上已经丧失了使用价值。

2.3　古典密码体制

古典密码体制采用手工或者机械操作实现加解密,相对简单,大多数古典加密早在计算机普及之前已经被开发出来,计算机出现后,由于计算机运行的速度远远高于手工计算速度,所有古典密码算法能够被计算机很容易地破解,目前任何重要的应用程序都不推荐使用古典加密算法。古典密码的安全性较弱,但反映了密码设计的一些基本原则和方法,回顾和研究这些密码的原理与技术,对于理解、设计和分析现代密码仍有较强的借鉴意义。

古典密码采用两种基本技术,即代换和置换技术,代换是将明文字母替换成其他字母、数字或符号。置换是打乱明文的字母位置形成密文。古典密码体制只使用代换或置换技巧,而现代密码体制大多是综合应用这两种技术实现的,相对于古典密码体制来说,基于的数学基础更加复杂。

2.3.1　代换密码

1. 恺撒密码

已知的最早的代换密码是由古罗马 Julius Caesar 发明的恺撒密码,这种密码将明文中每个字母用字母表中在它之后的第三个字母进行循环代换,恺撒密码的密码表如表 2.1 所示。

表 2.1 恺撒密码的密码表

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

【例 2-1】 用恺撒密码对 attack on three am 进行加密,得到的密文为 DWWDFN RQ WKUHH DP。

如果让每个字母等价于一个数值,如表 2.2 所示,那么恺撒密码的加解密公式分别为:

加密:$C_i=(P_i+3) \bmod 26$

解密:$P_i=(C_i-3) \bmod 26$

如果移位可以是任意整数 k,则更加通用的密码算法(也称移位代换密码)如下:

加密:$C_i=(P_i+k) \bmod 26$

解密:$P_i=(C_i-k) \bmod 26, k\in[0,\cdots,25]$

表 2.2 字母和数字对应关系表

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

由于密钥 k 只有 26 种可能的取值,因此用穷举分析可以轻松破解移位代换密码。

2. 单表代换密码

移位代换密码仅有 26 种可能的密钥,是很不安全的,之所以只有 26 个密钥是因为移位代换密码中的字母代换太有规律了,如果打破规律代换而允许任意代换,则密钥空间将会急剧增大,例如如果明文字母 A 用 C 代换,在移位代换中字母 B 只能用 D 代换,字母 C 只能用 E 代换……,以此类推,而如果采用任意代换则 B 可用除 C 之外剩余 25 个字母中的随机一个来代换,C 用剩余 24 个字母中的随机一个来代换……,以此类推,这样,密钥空间为 26!,约 4×10^{26} 种可能的密钥,这么大的密钥空间即使对于计算机来说也应该可以抵挡穷举攻击。因为每条消息用一个字母映射表(从明文字母到密文字母的映射)加密,所以这种方法称为单表代换密码。

【例 2-2】 采用单表代换加密的密码表如表 2.3 所示,对 attack on three am 进行加密,得到的密文为 FXXFGP ZQ XALMM FC。

表 2.3 单表代换加密的密码表

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	T	G	S	M	O	N	A	Y	V	P	D	C	Q	Z	R	W	L	E	X	H	B	K	I	U	J

攻击单表代换密码的有效方法是利用统计分析法,即利用语言规律进行攻击。公元 9 世纪,阿拉伯的密码破译专家就已经娴熟地掌握了用统计字母出现频率的方法来攻击单表代换密码。破解的原理很简单:在每种拼音文字语言中,每个字母出现的频率并不相同,例如在英语中,e 出现的次数要大大高于其他字母,英文字母使用频率分布如图 2.2 所示。所以如果取得了足够多的密文,通过统计每个字母出现的频率,就可以猜出密码中的一个字母对应于明文中的哪个字母(当然还要通过揣摩上下文等基本密码破译手段)。

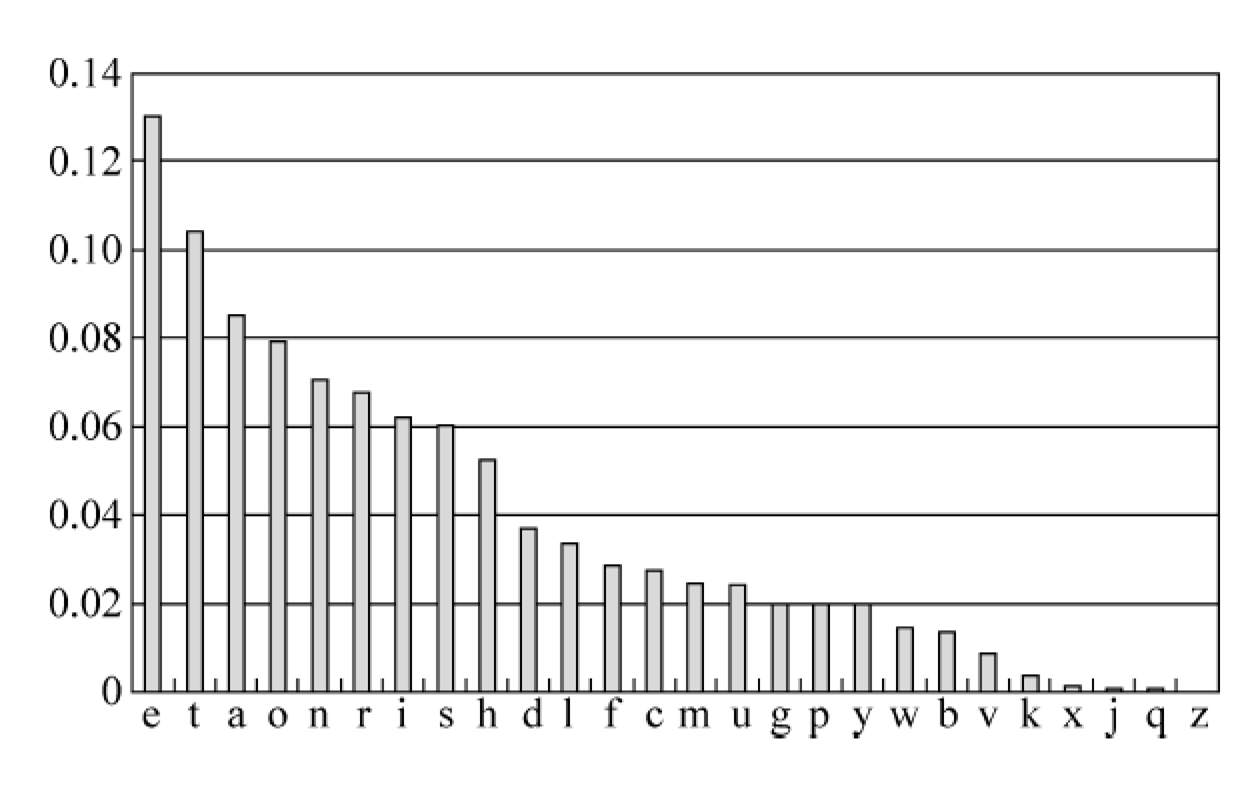


图 2.2　英语字母的统计规律图

如果消息足够长，只要用这种方法就足够了，但是如果消息比较短，还可以用到一些统计规律：

（1）英文单词以 E、S、D、T 结尾的超过一半。

（2）英文单词约以 T、A、S、W 为起始字母的一半。

（3）一般来说三个字母出现的可能是 THE 或 AND。

（4）单个字母出现的可能是 A 或 I。

（5）最常见的两字母组合，依照出现次数递减的顺序排列依次为：TH、HE、IN、ER、AN、RE、DE、ON、ES、ST、EN、AT、TO、NT、HA、ND、OU、EA、NG、AS、OR、TI、IS、ET、IT、AR、TE、SE、HI、OF。

单表代换之所以能采用统计的方法进行攻击，主要的原因是单表代换中每个明文字母只用一个密文字母来代替，明文中字母出现的频率信息还会保留在密文中。如果一个明文字母可以用多个字母代换，例如字母 A，有时用 B 代换，有时用 F 代换，可以使得明文字母出现的频率信息得到隐藏，这也是多表代换的思想。

3. 多表代换密码

改进单表代换的方法是在明文消息中采用多个单表代换，这样密文中的每个字母都有多个可能的密文字母来代换它，从而在密文中隐藏明文字母出现的频率信息，这种方法称为多表代换。Vigenere 密码是最为著名的多表代换密码。

Vigenere 密码是一种以移位代换为基础的周期代换密码，采用该算法，明文中每个字母采用移位代换法，所移位的位数由密钥决定。每一个密钥字母加密一个明文字母，直到所有的密钥字母用完，然后再从头开始，也就是密钥循环使用。

【例 2-3】 密钥词为 deceptive，那么明文 we are discovered save yourself 的加密过程如表 2.4 所示。

表 2.4　多表代换过程

w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	y	o	u	r	s	e	l	f
d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e
z	i	c	v	t	w	q	n	g	r	z	g	v	t	w	a	v	z	h	c	q	y	g	l	m	g	j

第一个明文字母 w(对应表 2.2 中的数字为 22),该明文字母采用移位代换进行加密,移位数目由对应的密钥字母 d(对应数字为 3)确定,则密文为 22+3(mod 26)=25,对应的密文字母为 z,其他密文字母的产生采用相同的方法。

采用 Vigenere 密码,例 2-3 中明文第 2 位和第 5 位的字母 e 对应的密文字母分别为 i 和 t,即每个明文字母对应多个密文字母,这样字母出现的频率信息就被隐蔽了。

Vigenere 密码如何进行破译呢?在上述例子中,明文中出现了两次 red,通过观察发现其对应的密文序列 vtw 也相同,这是因为两个 red 对应的密钥字母序列也相同。采用 Vigenere 密码加密,如果两个相同的明文序列之间的距离是密钥词长度的整数倍,那么产生的密文序列也是相同的。这样,密码分析者只要发现重复序列 vtw,且重复序列之间相隔 9 个字母,那么他就可以认为密钥词长度是 3 或者 9。vtw 的两次出现可能是偶然的,不一定是用相同密钥词加密相同明文所导致的,然而,如果信息足够长,就会有大量重复的密文序列出现,通过计算重复密文序列间距的公因子,分析者就能猜出密钥的长度。

采用 Vigenere 密码加密,间隔长度是密钥长度整数倍位置上的加密实际上是单表代换。如果密钥的长度为 N,那么加密过程包含了 N 个单表代换。例如例 2-3 中,位置 1,10,19,…的字母的加密是单表代换,第 2,11,19,…位的字母加密也是单表代换,以此类推,在分析出密钥长度后,对于每一个单表代换可以用明文语言的频率特性进行分析从而破解 Vigenere 密码。

虽然破译 Vigenere 密码的技术并不复杂,但是在 1917 年的一期《科学美国人》的杂志上却称之为不可破译的。当对现代密码算法做出类似论断时,这是值得吸取的教训。

Vigenere 密码之所以能被破解的主要原因是密钥重复使用,因此仍然能够用统计的方法进行分析,要抗击这样的密码分析,只有选择与明文长度相同的密钥。

4. 一次一密

Major Joseph Mauborgne 和 AT&T 公司的 Gilbert Vernam 于 1917 年发明了一种理想的加密方案,称为一次一密乱码本(One-time Pad),被认为是一种不可攻破的密码体制。一次一密乱码本是一个大的不重复的随机密钥字母集,每个密钥被分别写在一张纸上,所有密钥纸被粘成一个乱码本,发送者在发送消息时用乱码本中的一个密钥加密,然后销毁乱码本中用过的一页,接收者有一个相同的乱码本,依次用乱码本上的密钥解密,并销毁密钥本中用过的一页。采用这种方式,每个密钥仅对一个消息使用一次。

一次一密要求使用与消息本身一样长的随机密钥,每个密钥只能使用一次,一次一密的安全性完全取决于密钥的随机性。如果构成密钥的字符流是真正随机的,那么构成密文的字符流也是真正随机的,这样的密码是无条件安全的。

【例 2-4】 1918 年美国电报电话公司的 G. W. Vernam 提出这样的密码系统:明文英文字母编成 5bit 二元数字,称为五单元波多码(Baudot Code),选择随机二元数字流作为密钥,加密通过执行明文和密钥的逐位异或操作产生密文,可以简单地表示为:

$$C_i = P_i \oplus K_i$$

其中,P_i 表示明文的第 i 个二元数字,K_i 表示密钥的第 i 个二元数字,C_i 表示密文的第 i 个二元数字,⊕表示异或操作。解密仅需要执行相同的逐位异或操作:

$$P_i = C_i \oplus K_i$$

但实际上一次一密要达到无条件安全,存在两个基本难点:一是产生大规模随机密钥

的实际困难，另一个是密钥分配和保护问题，对每一条发送的消息，需要提供给发送方和接收方等长度的密钥，因此存在庞大的密钥分配问题，所以一次一密在实际中很少使用，而主要用于安全性要求很高的低带宽通信。

前苏联曾经在第二次世界大战后使用过一次一密的方法来加密间谍发送的消息，用一叠在每一页上都标有随机数的纸，每页纸用于一条消息，而且只用一次。如果正确使用，这种加密机制无法破解，但是苏联人的错误是没有正确使用它们，重复使用了一次性便条，所以一些消息就被破解了。

那么什么样的密码才是真正的一次一密呢？它必须满足三个条件：

(1) 密钥是随机产生的，而且必须是真随机数，而不是伪随机数。

(2) 密钥不能重复使用。

(3) 密钥的有效长度不小于密文长度。

2.3.2 置换密码

与代换不同的另外一种加密方式是打乱明文字母的顺序形成新的序列，这种技术称为置换密码。置换密码的基本思想是按一定的规则书写明文，而按另一规则读出密文。

早在公元前 6 年古希腊人借助于一根叫 scytale 的棍子进行加密，送信人将一张纸条环绕在 scytale 棍子上，把要加密的消息沿着棍子横写，将缠绕在棍子上的纸条展开后，纸条上的字母看起来是一些随机字母，如果不知道棍子的宽度(这里作为密钥)是很难解密的。这种加密方法就是典型的置换法。

典型的置换还可采用栅栏技术，这种加密方法按照对角线的顺序写入明文，而按行的顺序读出作为密文。

【例 2-5】 用深度为 2 的栅栏技术加密明文 meet after the toga party，其过程如表 2.5 所示，则密文为 MEATRHTGPRYETFETEOAAT。

表 2.5　栅栏技术加密过程

m	e	a	t	r	h	t	g	p	r	y
e	t	f	e	t	e	o	a	a	t	

一种更加复杂的方案是把消息一行一行地写成矩形块，然后按列读出，但是把列的次序打乱，列的次序就是密钥。

【例 2-6】 用矩阵法加密明文 meet after the toga party，如表 2.6 所示。

表 2.6　用矩阵法加密明文

密钥	4	3	1	2	5	6	7
明文	m	e	e	t	a	f	t
	e	r	t	h	e	t	o
	g	a	p	a	r	t	y

则生成的密文为 ETPTHAERAMEGAERFTTTOY，如表 2.7 所示。

解密时将密文分组后按列的顺序排列，并根据密钥重新排列列的顺序，原来的第四列调整到第一列、第三列调整到第二列，以此类推，最终得到表 2.6，按行的顺序读出即可得到明

文序列。

表 2.7 用矩阵法解密密文

密文	e	t	e	m	a	f	t
	t	h	r	e	e	t	o
	p	a	a	g	r	t	y

单纯的置换技术对于现代密码分析来说是微不足道的。因此，置换技术通常是与代换技术相结合使用的，一般可先用代换技术加密，再用置换技术将密文再次加密。

2.4 对称密码体制

相对于古典密码体制，现代密码体制的算法是针对比特而不是针对字母进行变换，并且算法更加复杂，但是采用的技术还是没变，大多数优秀算法的主要组成部分仍然是代换和置换的组合。现代密码体制按密钥特征进行划分，可以分为对称密码体制和公钥密码体制，其中对称密码体制是指加密和解密用到的密钥相同，或者存在确定的转换关系。根据密码算法对明文信息的加密方式，对称密码体制分为两类，即分组密码和序列密码(或称流密码)。其中分组密码是先把明文划分为长度相等的分组(分组大小通常为 64b 或 128b)，每个明文分组被当作一个整体来产生一个等长(通常情况下)的密文分组，分组密码体制是目前商业领域中比较重要而流行的一种加密机制，广泛应用于数据的保密传输、加密存储等应用场合。序列密码每次加密数据流中的一位或一个字节。目前流行的对称密码有 DES、3-DES、AES、IDES、Blowfish、RC 系列算法，这里主要介绍数据加密标准 DES 算法。

2.4.1 DES 简介

20 世纪 60 年代计算机应用得到了迅猛的发展，大量数据资料被集中存储在计算机数据库中并在计算机通信网中进行传输，有些通信内容具有高度机密性，例如大额度的转账信息、有价证券购买或出售信息、逮捕令、航班和票务预定、医疗和保险记录等，因此对计算机通信及数据进行保护的需求日益增长。

1973 年，美国国家标准局(NBS)发布密码算法征集通知，公开征求一种标准算法用于保护计算机数据的传递和存储。IBM 公司 Feistel 领导的设计小组提交了他们研制的一种密码算法，该算法是由早期的 LUCIFFR 密码改进而得的。在经过大量的公开讨论后该密码算法于 1977 年 1 月被正式批准为美国数据加密标准，此后，这一受到批准的算法被称为 DES(Data Encryption Standard)，1980 年 12 月美国国家标准协会 ANSI 正式采用该算法作为美国商用加密算法。

DES 设计巧妙，除了密钥输入顺序，其加密和解密的步骤完全相同，在 DES 出现后，经过许多专家学者的分析论证证明该算法是一种性能良好的数据加密算法，不仅随机特性好、线性复杂度高，而且易于实现，因此 DES 在国际上得到了广泛的应用，它的产生被认为是信息加密技术发展史上的里程碑之一。

2.4.2　DES 加解密原理

1. DES 的加密

DES 算法是典型的分组密码，加密前先将明文编码表示后的二进制序列划分成长度为 64b 的分组，对于每个分组执行如图 2.3 所示的算法步骤，DES 算法的密钥也是长度为 64b 的二进制序列，密钥中第 8、16、24、32、40、48、56、64 位为奇偶校验位，因此真正起作用的只有 56 位，算法的输出为 64b 的密文。

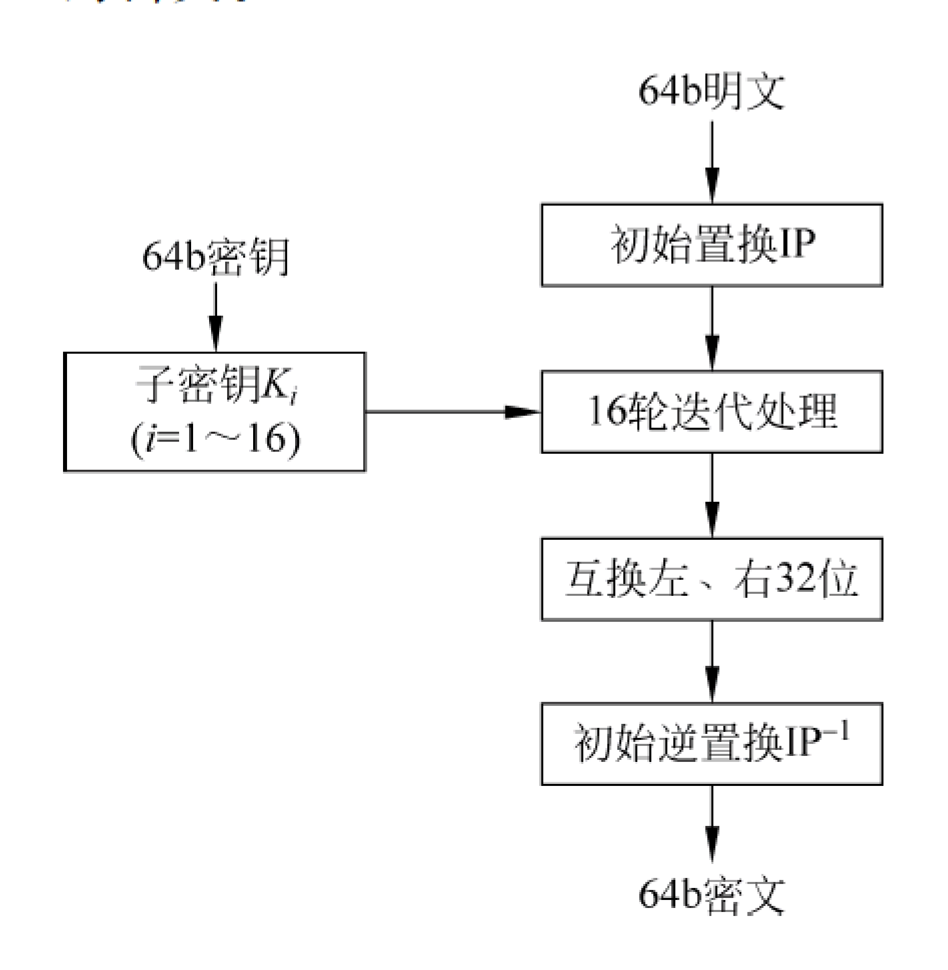


图 2.3　DES 算法步骤

下面具体介绍一个分组的加密过程。

1）初始置换 IP

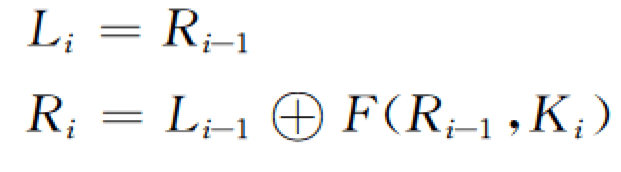
初始置换 IP 是将 64b 的明文进行位置重排，通过 IP 运算得到一个乱序的 64b 明文组，置换表如表 2.8 所示。置换后的数据平均分成左右两段，用 L 和 R 来表示，这两部分数据是下一步迭代变换的初始输入。

表 2.8　初始置换

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

2）迭代变换

它是 DES 算法的核心部分。如图 2.4 所示，将 1）中经过 IP 置换后的数据分成左右两组各 32b，作为第一轮迭代变换的输入。每轮迭代只对右边的 32b 进行一系列的加密变换 F，加密变换具体包括选择运算 E、密钥加密运算、选择压缩运算 S、置换运算 P。在一轮迭代即将结束时，把上一轮左边的 32b 与本轮经加密变换 F 得到的 32b 进行模 2 相加，作为下一轮迭代时右边的段，并将上一轮右边的未经变换的段直接送到左边的寄存器中作为下一轮迭代时左边的段，即：



$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

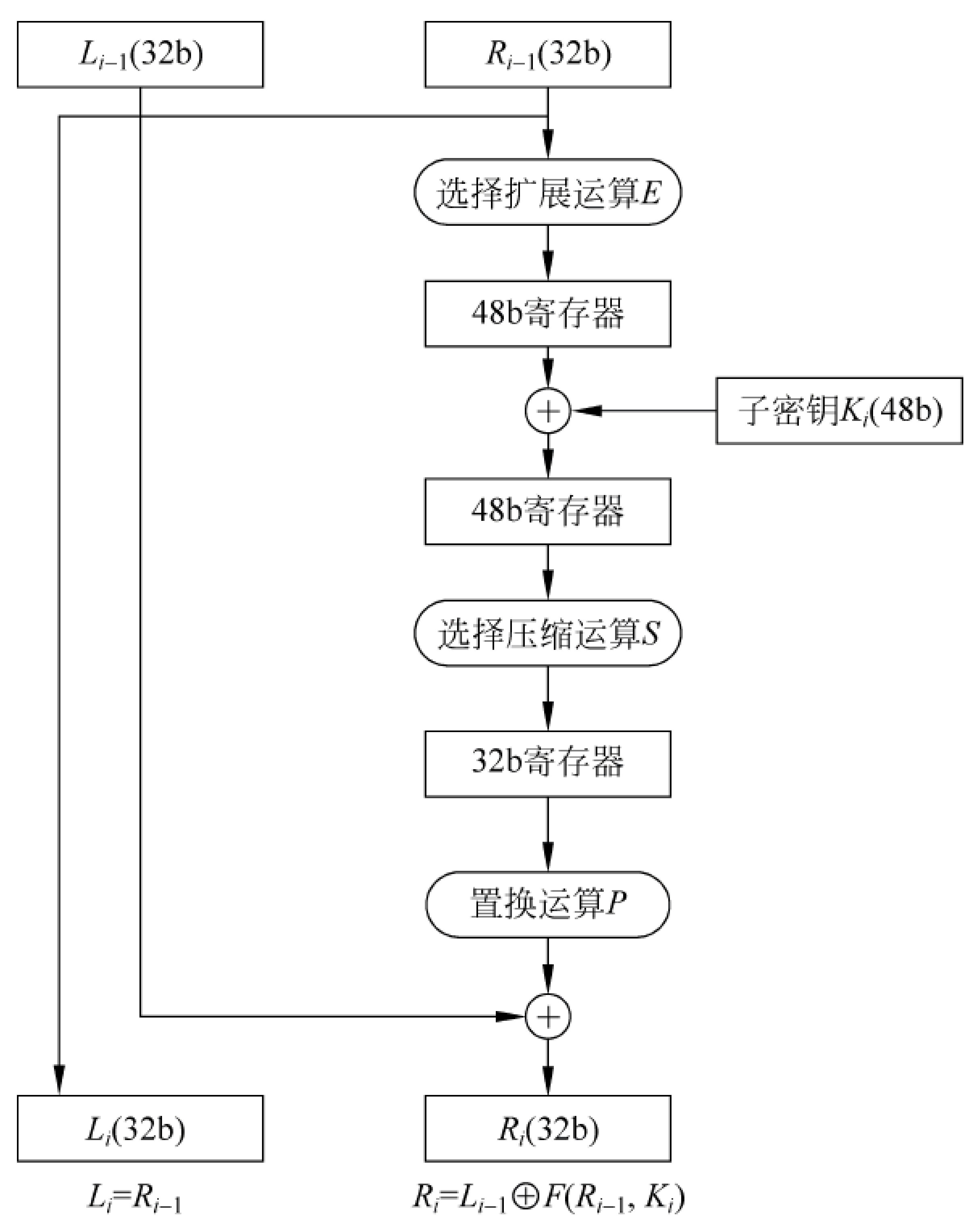


图 2.4 迭代变换

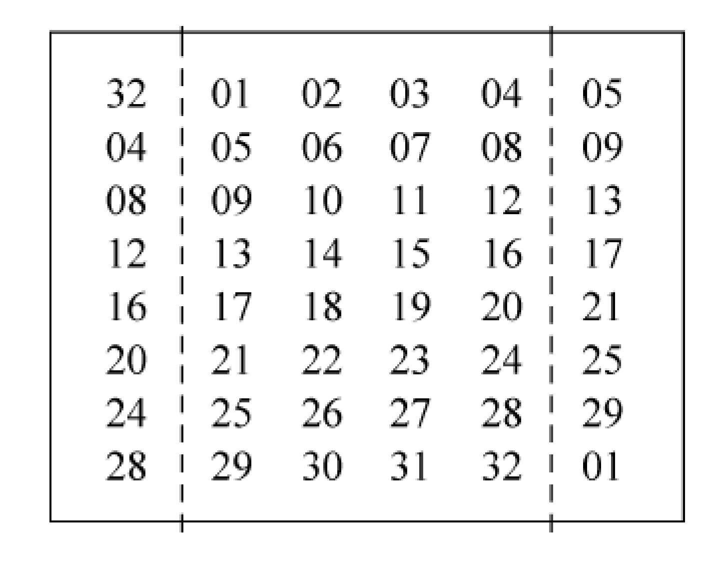
这样的迭代共进行 16 轮，结束后，再将所得的左、右长度相等的 L_{16} 和 R_{16} 进行交换得到 64b 数据。下面具体介绍加密变换，包括选择扩展运算 E、密钥加密运算、选择压缩运算 S、置换运算 P。

(1) 选择扩展运算 E

选择扩展运算将输入的 32b 扩展成 48b 输出，扩展方法是重复某些位置上的元素，其变换表如图 2.5 所示，共有 16 个位置的元素被读了两次。

(2) 密钥加密运算

将子密钥产生器输出的 48b 子密钥 K 与选择扩展运算 E 输出的 48b 数据按位模 2 相加(子密钥如何产生请见后文)。



32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

图 2.5 选择扩展运算

(3) 选择压缩运算 S

将(2)中产生的 48 比特数据自左至右分成 8 组，每组 6b，然后并行送入 8 个 S 盒。每个 S 盒为一非线性代换网络，能够将 6b 的输入转换为 4b 的输出，具体做法是对于每个 S 盒，将输入的第 1 和第 6 位组成的二进制数作为横坐标，其他 4b 作为纵坐标，然后查询相应的 S 盒，再将对应位置上的十进制数用二进制表示。盒 S1～S8 的选择函数关系如表 2.9 所示，运算 S 的框图如图 2.6 所示。

表 2.9　DES 的 S 盒定义

S1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	1	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	1	10	7	13	15	12	9	0	3	5	6	11

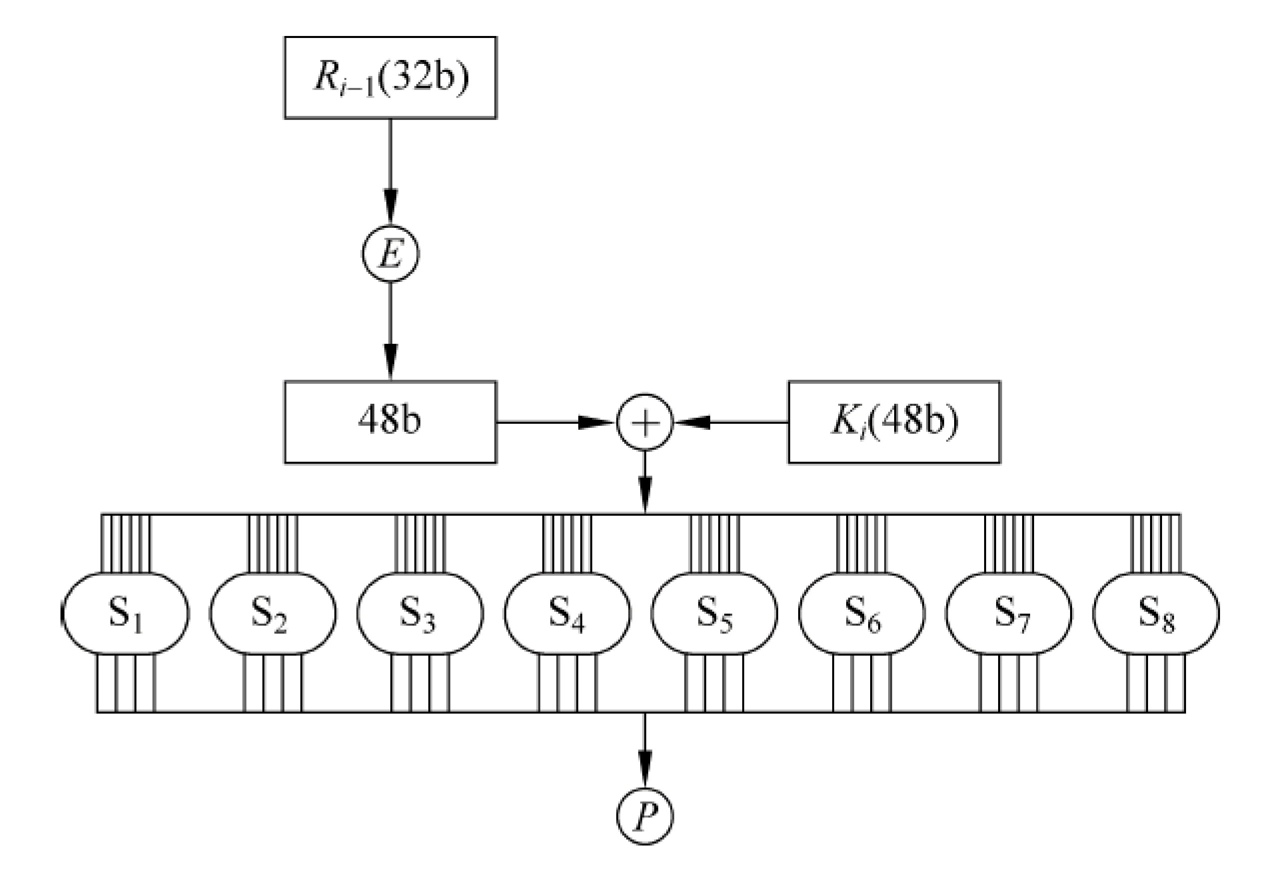


图 2.6　选择压缩运算过程

【例 2-7】 若对于 S_6 的输入为 110011，则行号为 $11_2=3$，列号为 $1001_2=9$，查询 S_6 的第 3 行第 9 列得到的十进制数为 14，转换为二进制数 1100，所以输出为 1100。

(4) 置换运算

置换运算 P 对 $S_1 \sim S_8$ 盒输出的 32b 数据进行坐标变换，具体置换方法如表 2.10 所示。

表 2.10 置换 P 表

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

3) 子密钥产生器

DES 加密过程共涉及 16 轮迭代，每轮使用一个不同的 48 位子密钥，共需 16 个子密钥，这些子密钥由初始输入的 64 位密钥产生，初始密钥中有 8 位为奇偶校验位，位置号分别为 8、16、24、32、40、48、56 和 64，因此真正有效的只有 56 位，子密钥的具体生成过程如图 2.7 所示。

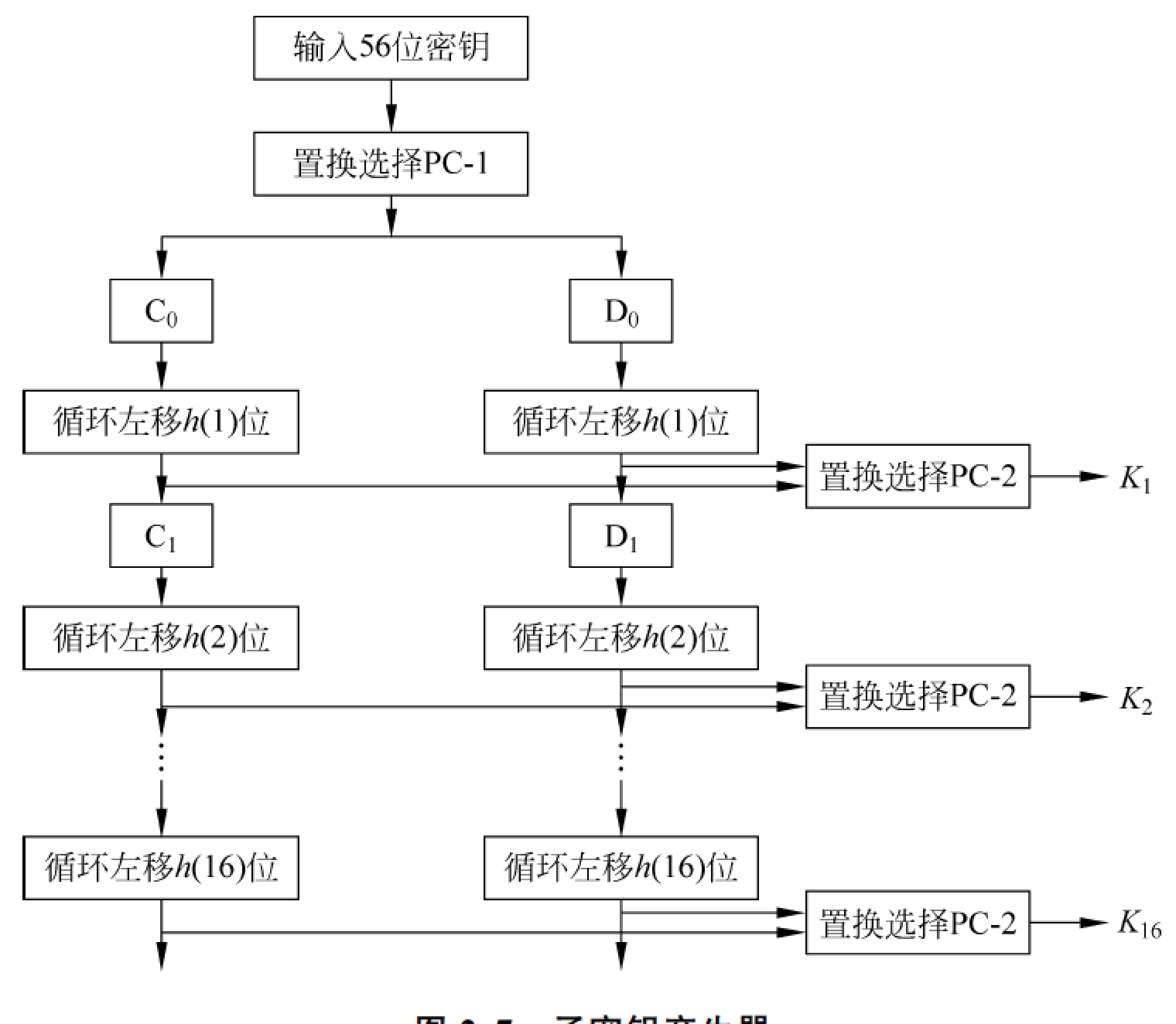


图 2.7 子密钥产生器

56 位密钥首先经过置换选择 PC-1(如表 2.11 所示)，将其位置打乱重排，置换后分为两组，每组为 28b，分别送入 C 寄存器和 D 寄存器中，接下来对 C 和 D 寄存器中的数据进行左循环移位置换，每轮移位数目如表 2.12 所示，移位后将 C 和 D 寄存器的存数送给置换选择 PC-2，从中挑出 48b 作为这一轮的子密钥，这个子密钥作为前面介绍的加密函数的一个输入，再将 C 和 D 寄存器的存数循环左移后，使用置换选择 PC-2 产生下一轮迭代的子密钥，如此继续，产生所有 16 个子密钥。

表 2.11　置换选择 PC-1

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

表 2.12　移位次数表

第 i 次迭代	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
循环左移次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

置换选择 PC-2(见表 2.13)将 C 中第 9、18、22、25 位和 D 中的第 7、9、15、26 位删除,并将其余数字置换位置后送出 48b 数字作为第 i 次迭代时所用的子密钥 K_i。

表 2.13　置换选择 PC-2

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

4) 逆初始置换 IP^{-1}

如表 2.14 所示,逆初始置换 IP^{-1}将 16 轮迭代后给出的 64b 组进行置换得到输出的密文组,逆初始置换后得到的 64b 数据分组,即为加密后得到的密文。

表 2.14　逆初始置换

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

2. DES 算法的解密

解密算法与加密算法相同,只是子密钥的使用次序相反。把 64 位密文当作输入,第一次解密迭代使用子密钥 K_{16},第二次解密迭代使用子密钥 K_{15},…,第 16 次解密迭代使用子密钥 K_1,最后输出的便是 64 位的明文。

3. DES 算法的工作模式

DES 是对称分组密码体制,分组长度为 56,但实际要加密的消息不一定刚好是一个分组长度,为了能在实际中应用,分组密码可以在不同的操作模式下运行,允许用户选择不同的模式满足他们的应用需求,有 5 种常见的操作模式:电子密码本(ECB)、密码分组链接(CBC)、密码反馈(CFB)、输出反馈(OFB)和计数器(CTR)模式,这里主要介绍 ECB 和 CBC 模式。

1) 电子密码本(Electronic Code Book,ECB)

这种方式是分组密码的基本工作方式,它将长的明文分成大小相等的分组 $P=(P_1,P_2,\cdots,P_L)$,最后一组在必要时需要进行填充,每组用相同的密钥 K 进行加密 $C_j=$

$E_K(P_j)$，加密后将各组密文合并成密文消息 $C=(C_1,C_2,\cdots,C_L)$，如图 2.8 所示。

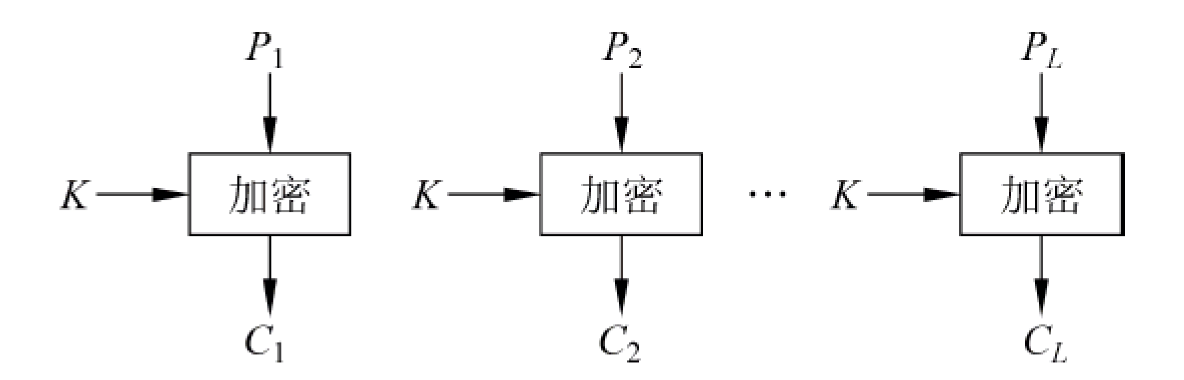


图 2.8　电子密码本方式

在 ECB 模式下，每一个分组独立加密，产生独立的密文组，采用这种方式可以利用并行处理来加速加密运算和解密运算，并且在传输时任意一个分组的错误不会影响其他分组，这是该模式的一个优点。但是，相同的明文组产生相同的密文组，当处理长的明文时，ECB 工作模式就会暴露出弱点。假设敌方 Eve 在充分长的一段时间内观察了 Alice 和 Bob 之间的通信，如果 Eve 已经设法获得了一些密文对应的明文，就可以开始建立一个密码本用来解密 Alice 和 Bob 之间未来的通信信息，Eve 没有必要计算 K，只要查询一下密码本中的密文所对应的明文来解密消息。

2）密码分组链接(CBC)

为了克服 ECB 的缺陷，人们希望设计一种方案使同一明文分组重复出现时产生的密文分组不同。一种简单的方案是使用密码分组链接(Cipher Block Chaining，CBC)模式，如图 2.9 所示，这种模式和 ECB 模式一样，也要将明文分成大小相等的分组 $P=(P_1,P_2,\cdots,P_L)$，最后一组在必要时需要进行填充，CBC 将这些分组连接在一起进行加密，加密输入是当前明文分组和前一密文分组的异或，它们形成一条链，每次加密使用相同的密钥。在加密时，最开始一个分组先和一个初始向量(Initialization Vector，IV)进行异或，然后用密钥加密，每个分组的加密结果均会受到前面所有分组的影响，所以即使相同的明文分组也会产生不同的密文，有利于保护明文。但是 CBC 模式会导致错误传播，密文传输中任何一组发生错误不仅影响该分组的正确解密，也会影响其下一分组的正确解密。该模式的另一个缺点是不能实时解密，也就是说必须等到所有分组密文收到之后才能解密。

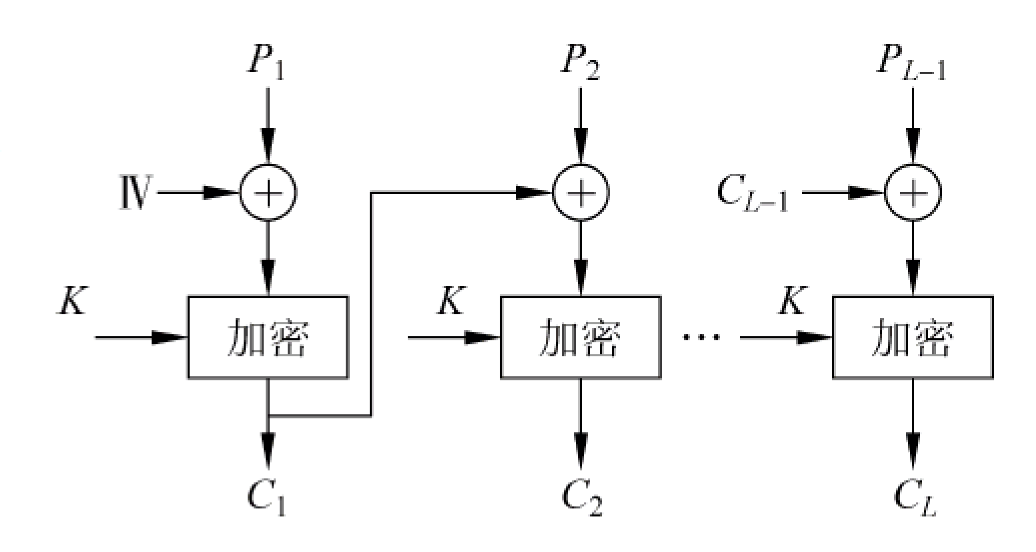


图 2.9　密码分组链接方式

2.4.3　DES 的安全性

DES 的出现是密码学史上的一个创举，在此之前密码体制及其设计细节都是严加保密的，而 DES 算法公开发表，可供任何人研究和分析，DES 的安全性完全依赖于密钥。DES 在 20 多年的应用实践中，没有发现严重的安全缺陷，在世界范围内得到了广泛的应用，为确保信息安全做出了巨大贡献。

从应用实践来看，DES 具有良好的“雪崩效应”。所谓“雪崩效应”是指明文或密钥的微小改变将对密文产生很大影响。DES 显示了很强的雪崩效应，在密钥相同的情况下，明文的一位变化，3 轮迭代后密文有 21b 不同，16 轮迭代后有 34b 不同。

当 DES 算法被建议作为一个标准时，曾出现过很多批评。其中最有争议的问题之一就

是 S 盒。S 盒的设计原则是 DES 的安全核心，因为在 DES 算法中，除了 S 盒外，所有计算都是线性的。S 盒的设计被列为官方机密，所以有人认为 S 盒可能存在陷门，美国国家安全局(NSA)有可能利用这些弱点在没有密钥的情况下解密，但至今没有迹象表明 S 盒中存在陷门。IBM 在 20 世纪 90 年代早期公布了如下设计准则。

(1) 每一个 S 盒的输入是 6 位，输出是 4 位。这是 1974 年在一个芯片上的能放大最大的内容。

(2) S 盒的输出不应该和输入的线性函数接近(线性性质可以使系统更容易分析)。

(3) S 盒的每一行包含 0～15 的所有数。

(4) 如果 S 盒的两个输入只有 1 位不同，那么这两个输出至少有 2 位不同。

(5) 如果 S 盒的两个输入的前 2 位不同，后 2 位相同，那么输出一定不同。

(6) 给定 32 对输入的异或，对每一对计算输出的异或，至多有 8 个相同。这显然是为了抵御差分攻击。

关于 DES 算法另一个具有争议的问题就是担心实际 56b 的密钥长度不足以抵御穷举攻击，因为密钥量只有 2^{56}。1977 年 Differ 和 Hellman 认为利用 100 万个超大规模集成电路块所组成的一台专门用于破译 DES 的并行计算机能在一天中穷举搜索所有 2^{56} 个密钥，这样的一台计算机在 1977 年需要耗资 2000 万美元，Differ 和 Hellman 指出，除了像美国国家安全局那样的机构，任何人不可能破译 DES，但他们预测到 1990 年制造和破译 DES 专用机的成本将大幅度下降，那时 DES 将完全不安全。

事实证明，他们的预测是有道理的，随着计算能力和 Internet 网络的发展，DES 已经不能经受住穷举攻击。1997 年 1 月 28 日，美国的 RSA 数据安全公司在 RSA 安全年会上悬赏 10 000 美金破解 DES，科罗拉多州的程序员 Verser 在 Internet 上数万名志愿者的协作下用 96 天的时间找到了 DES 密钥。1998 年 7 月电子前沿基金会(EFF)使用一台价值 25 万美元的计算机在 56 小时之内破译了 56b 的 DES，1999 年 1 月电子前沿基金会(EFF)通过互联网上的 10 万台计算机合作，仅用 22 小时 15 分就破解了 56b 的 DES。这一事件表明，依靠 Internet 的分布式计算能力，用穷举攻击方法破解 DES 已经成为可能，因此需要寻找新的算法代替 DES 算法。

2.4.4　三重 DES

DES 由于安全问题，美国政府于 1998 年 12 月宣布 DES 不再作为联邦加密标准。在新的加密标准实施前，为了使已有的 DES 算法投资不浪费，人们尝试用 DES 和多个密钥进行多次加密，其中三重 DES 已被广泛采用。

1. 二重 DES

最简单的多重 DES 加密是用 DES 加密两次，每次使用不同的密钥，二重 DES 的加密与解密过程如图 2.10 所示，给定明文 P 和两个加密密钥 k_1 和 k_2，二重 DES 加密过程为 $C=E_{k_2}(E_{k_1}(P))$，解密过程为 $P=D_{k_1}(D_{k_2}(C))$，因为使用了两个 64 位的密钥，所以二重 DES 的密钥总长度为 112 位，密钥空间的数量为 2^{112}，似乎密码强度增加了一倍，但是如果采用“中间相遇攻击”(Meet-in-the-Middle Attack)来攻击，则可以大大减少攻击代价。

从图 2.10 中可以看出：$X=E_{k_1}(P)=D_{k_2}(C)$。

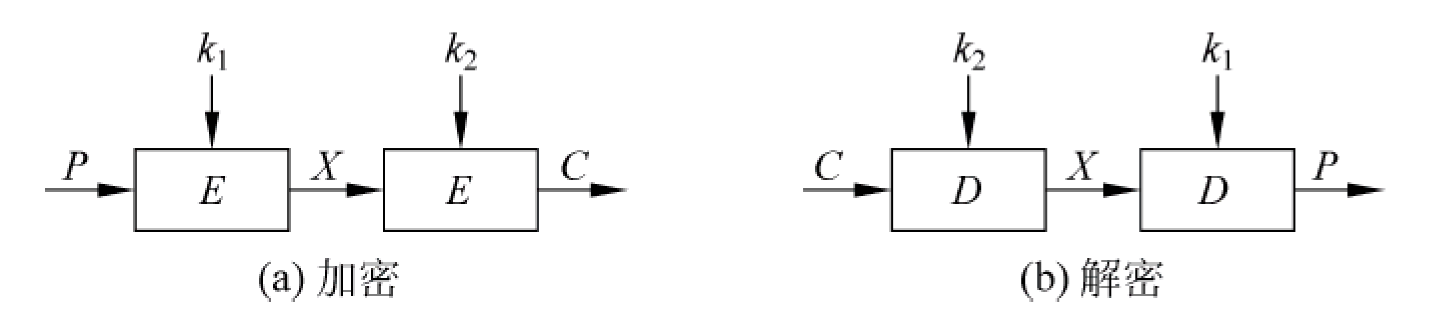


图 2.10　两重 DES 算法过程

若给出一个已知的明-密文对(P,C),分别用 2^{56} 个密钥 k_1 对明文进行加密,得到一张密钥/密文 X 对应表,类似的,用 2^{56} 个密钥 k_2 对密文进行解密,得到相应的明文 X 对应表,比较两个表中 X 相同的项,就会得到真正实用的密钥对 k_1、k_2,可以看出计算代价为 $2^{56}+2^{56}=2^{57}$。

2. 三重 DES

为了防止中间相遇攻击,可以采用三次加密方式,如图 2.11(a)所示。这是使用两个密钥的三重 DES,采用加密-解密-加密(EDE)方案。加密为 $C=E_{k_1}(D_{k_2}(E_{k_1}(P)))$,解密过程为 $P=D_{k_1}(E_{k_2}D_{k_1}(C))$,如图 2.11(b)所示,这种加密方案的攻击代价为 2^{112}。

目前还没有针对两个密钥的三重 DES 的实际攻击方法,但是感觉它不太可靠,又建议使用三密钥的三重 DES,加密为 $C=E_{k3}(D_{k2}(E_{k1}(P)))$,解密过程为 $P=D_{k_1}(E_{k_2}D_{k_3}(C))$,此时密钥长度为 168b。目前这种加密方式已经被一些网络应用采用,例如 PGP 和 S/MIME 就采用了这种方案。

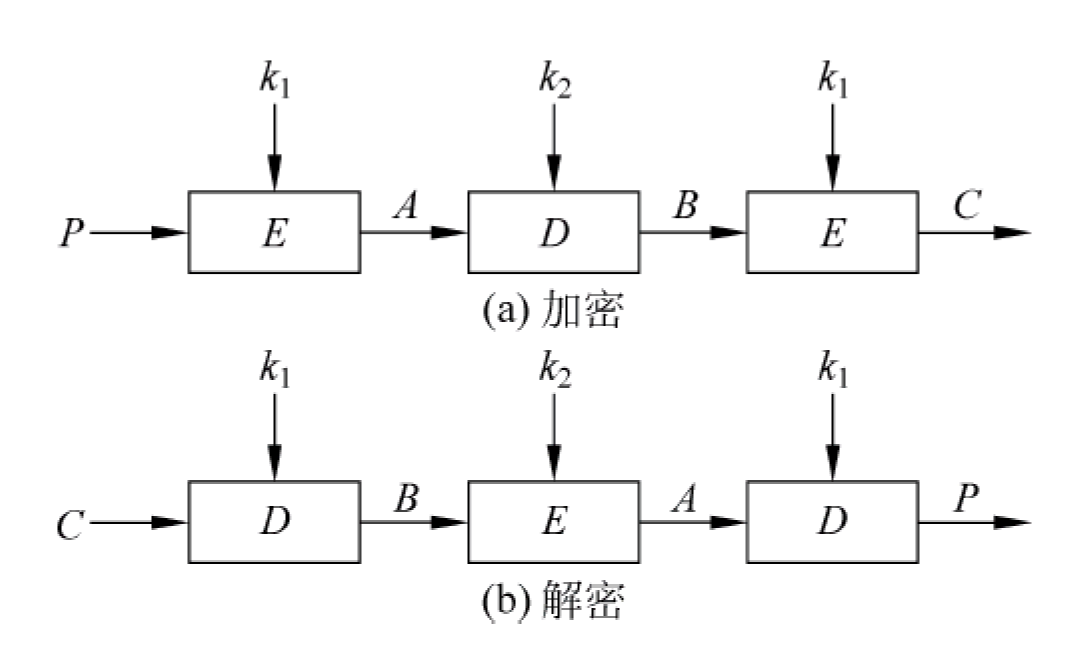


图 2.11　三重 DES 算法过程

2.4.5　高级加密标准 AES

AES 是美国国家标准技术研究所 NIST 旨在取代 DES 的新一代的加密标准。NIST 对 AES 候选算法的基本要求是：对称分组密码体制；密钥长度支持 128、192、256 位；明文分组长度 128 位；算法应易于各种硬件和软件实现。1998 年 NIST 开始 AES 第一轮征集、分析、测试,共产生了 15 个候选算法。1999 年 3 月完成了第二轮 AES 的分析、测试。1999 年 8 月 NIST 公布了 5 种算法(MARS、RC6、Rijndael、Serpent、Twofish)成为候选算法。最后,Rijndael,这个由比利时人设计的算法与其他候选算法在为高级加密标准(AES)的竞争中取得成功,于 2000 年 10 月被 NIST 宣布成为取代 DES 的新一代的数据加密标准,即 AES。尽管人们对 AES 还有不同的看法,但总体来说,Rijndael 作为新一代的数据加密标准汇聚了强安全性、高性能、高效率、易用和灵活等优点。AES 设计有三个密钥长度：128b、192b、256b,相对而言,AES 的 128b 密钥比 DES 的 56b 密钥强 10^{21} 倍。

AES 的 128 位输入可以看成一个 4×4 矩阵 S,这个矩阵称为“状态”(State)。例如,假设输入为 16 个字节 $b_0,b_1,\cdots,b_{15}$,这些字节在状态中的位置及其用矩阵的表示如表 2.15 所示。

表 2.15　AES 中状态的位置及其矩阵表示

(a)

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

(b)

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

AES 算法属于分组密码算法，它的输入分组、输出分组以及加/解密过程中的中间分组都是 128 位。本书用 N_r 表示对一个数据分组加密的轮数，N_r 依赖于密钥长度，密钥长度为 128 位，$N_r=10$；密钥长度为 192 位，$N_r=12$；密钥长度为 256 位，$N_r=14$。下面以分组长度为 128 位、密钥长度为 128 位，即加密轮数 10 为例，介绍 AES 算法加密与解密的过程。

1）加密过程

如图 2.12 所示，加密算法的 4 个步骤如下。

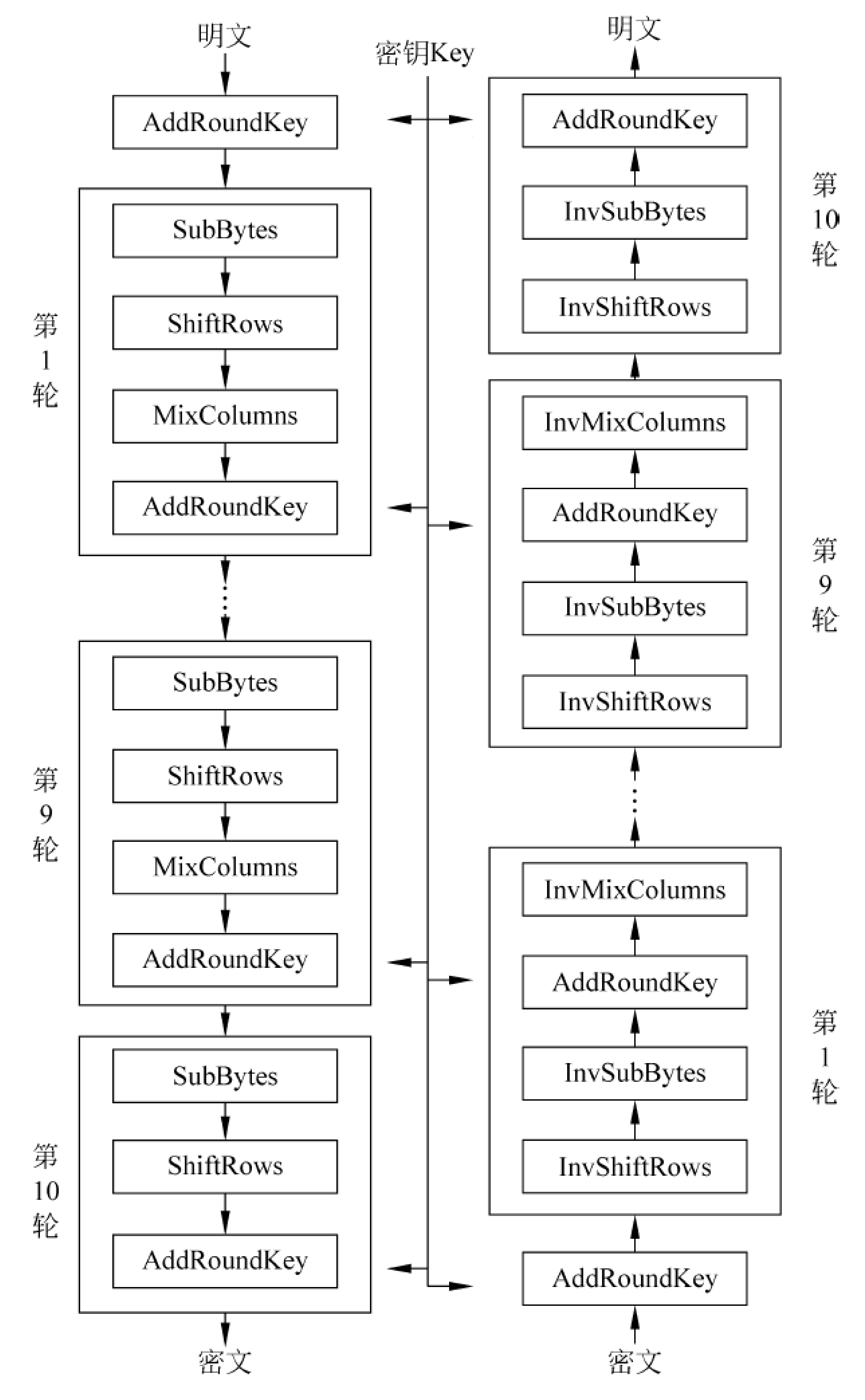


图 2.12　AES 算法加密和解密流程

(1) 给定一个明文 x,将 State 初始化为 x,并进行轮密钥(AddRoundKey)操作,该操作是将轮密钥与 State 进行异或。

(2) 执行 N_r-1 轮操作,每轮涉及如下操作步骤:对当前的 State 进行 S 盒变换操作(SubBytes)、行移位(ShiftRows)、列混淆操作(MixColumns)以及轮密钥(AddRoundKey)操作。

(3) 在最后一轮,对当前的 State 进行 SubBytes、ShiftRows、AddRoundKey 操作。

(4) State 中的内容即为密文。

上述加密过程涉及了 5 个重要操作:

(1) AddRoundKey—轮密钥加变换操作。将输入或状态 State 中的每一个字节分别与产生的密钥的每一个字节进行异或操作。

(2) SubByte—S 盒变换操作。SubByte 操作是一个基于 S 盒的非线性置换,S 盒是一个 16 行 16 列的矩阵,矩阵中每个元素为一个字节,如表 2.16 所示。将 State 状态中的每一个字节通过查表操作映射成另一个字节。映射方法是:输入字节的高 4 位作为 S 盒的行值、低 4 位作为 S 盒的列值,然后取出 S 盒中对应的行和列的值作为输出。例如,输入为 11000100 时,行值为 c,列值为 4(十六进制),S 盒中相应位置上的值为“1c”,这样 11000100 就被映射成了 00011100。

表 2.16 S 盒变换(十六进制)

行＼列	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

(3) ShiftRows—行移位操作。行移位的原则是:中间状态矩阵 State 的第 0 行不动,第 1 行循环左移 1 个字节,第 2 行循环左移 2 个字节,第 3 行循环左移 3 个字节,如图 2.13 所示。

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

(a) S

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

(b) S'

图 2.13　ShiftRows 完成行移位操作

(4) MixColumns—列混合变换操作。对中间状态矩阵 State 逐列进行变换。

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix}$$

(5) 密钥扩展。根据加密的轮数用相应的扩展密钥的 4 个数据项和中间状态矩阵上的列进行按位异或。首先定义几个数组和操作。

① $w[i]$：存放生成的密钥。

② Rcon[i]：存放前 10 个轮常数 RC[i]的值(用十六进制表示)，见表 2.17。其对应的 Rcon[i]见表 2.18。Rcon[i]=(RC[i],‘00’,‘00’,‘00’),RC[0]=‘01’,RC[i]=2RC[$i-1$]。

③ RotWord()操作：循环左移 1 个字节，将($b_0b_1b_2b_3$)变成($b_1b_2b_3b_0$)。

④ SubWord()操作：基于 S 盒对输入字中的每个字节进行 S 代替。

表 2.17　RC[i]中的值

i	1	2	3	4	5	6	7	8	9	10
RC[i]	01	02	04	08	10	20	40	80	1b	36

表 2.18　Rcon[i]中的值

i	1	2	3	4	5
Rcon[i]	01000000	02000000	04000000	08000000	10000000
i	6	7	8	9	10
Rcon[i]	20000000	40000000	80000000	1b000000	36000000

AES 算法利用外部输入密钥 K，通过密钥扩展程序得到共 $4(N_r+1)$字的扩展密钥 $w[4\times(N_r+1)]$。密钥扩展涉及如下三个模块的具体步骤：

(1) 初始密钥直接被复制到数组 $w[i]$的前 4 个字节中，得到 $w[0]$、$w[1]$、$w[2]$、$w[3]$。

(2) 对 w 数组中下标不为 4 的倍数的元素，只是简单地异或，即：

$w[i]=w[i-1]\oplus w[i-4]$(i 不为 4 的倍数)。

(3) 对 w 数组中下标为 4 的倍数的元素，在使用上式进行异或前，需对 $w[i-1]$进行一系列处理，即依次进行 RotWord、SubWord 操作，再将得到的结果与 Rcon[$i/4$]进行异或运算。

2) 解密过程

如图 2.12 所示，基本运算中除轮密钥 AddRoundKey 操作不变外，其余操作，包括 S 盒

变换操作(SubBytes)、行移位(ShiftRows)、列混淆操作(MixColumns)都要求进行求逆变换,分别记作 InvSubBytes、InvShiftRows、InvMixColumns。

在不同的安全系统中,还存在着其他对称加密算法,其中包括:

(1) IDEA。国际数据加密算法(International Data Encryption Algorithm)是由旅居瑞士的华人来学嘉和他的导师 J. L. Massey 共同开发的。IDEA 使用 128 位密钥,明文和密文分组长度为 64 位,已被用在多种商业产品中。

(2) Blowfish。Blowfish 允许使用最长为 448 位的不同长度的密钥,并针对在 32 位处理器上的执行进行了优化。

(3) Twofish。Twofish 使用 128 位分组,可以使用 128、192 或 256 位密钥。

2.5　公钥密码体制

2.5.1　公钥密码体制的产生

对称密码体制在加密和解密时,使用的是同一密钥,或者虽然使用不同的密钥,但是能通过加密密钥方便地导出解密密钥,因此,加密密钥是整个密码通信系统的核心机密,一旦加密密钥被暴露,整个密码体制也就失去了安全保密作用。

随着信息加密技术的应用领域从单纯的军事、外交、情报领域,扩大到商业、金融、计算机通信网络中的信息保密等民用领域,对称密码体制在应用中暴露出越来越多的缺陷。

1. 密钥管理十分困难

密钥管理是对称密码体制遇到的最大困难之一。在以对称密码体制为基础的保密通信中,通信双方需要在通信开始前分配相同的密钥,当用户数量很大时,互相之间通信需要大量密钥。如在一个民用通信网中,如果用户数为 n,为了实现用户两两之间的保密通信,系统需要管理 $n(n-1)/2$ 个密钥,当 n 等于 1000 时,网络中共需 499 500 个不同的密钥,产生、保存、分配、管理如此大量的密钥,本身就是一个难题。

2. 密钥传递的安全性无法保障

对称密码体制由于加解密密钥相同,因此在传输任何密文之前,发送者和接收者必须使用一个安全信道预先传送密钥,受到经济条件的限制,每两个用户之间都建立专用的秘密信道是不可行的,因此保证密钥传输的绝对安全在实际应用中是很难做到的。

3. 无法提供不可否认性服务

如果采用对称密码机制进行商业往来信息的加密,由于消息的发送方和接收方均拥有相同的密钥,接收方完全有能力篡改接收到的文件或伪造文件,发送方也可以抵赖他曾发出文件,而第三方没有足够的证据分辨实情。

例如,用户甲利用商用对称密码通信向用户乙订购了一批货物,用户乙如数寄出,后来由于该商品价格猛跌,用户甲抵赖发送过订单,拒绝付款,这样甲乙双方就发生争端,由于采用对称密码体制,双方都拥有相同的密钥,因此加密的订单双方都能产生,用户乙不能提供有说服力的法律证据来证明该订单是用户甲产生并发送的,此案法院无法受理。

对称密码体制存在上述缺陷,人们希望能设计一种新的密码,从根本上克服对称密码体

制存在的问题,公钥密码体制的出现正好弥补了上述缺陷。1976 年美国斯坦福大学的 Differ 和 Hellman 发表了 *New Direction in Cryptography* 一文,第一次提出了公钥密码体制的思想,开创了密码学的新时代。公钥密码体制的出现是密码学发展史上的一次革命,从古老的手工密码,到机电式密码,直至运用计算机的现代对称密码,这些编码系统虽然越来越复杂,但都建立在基本的替换和置换工具的基础上,而公钥密码体制的编码系统是基于数学函数(例如单向陷门函数)的。由于公钥密码算法不需要联机密码服务,密钥分配协议简单,所以极大地简化了密钥管理,除了加密功能外,公钥密码还可以提供数字签名。

自 1976 年以来,已经提出了多种公钥密码算法,其安全基础都是基于一些在短期内不可能得到解决的数学难题,如整数因子的分解难题至今已有数千年的历史,可以认为基于这些数学难题的公钥密码体制是安全的。

2.5.2　公钥密码体制的基本原理

1. 公钥密码体制的基本构成

在公钥密码体制中密钥是成对出现的,一个为加密密钥,一个为解密密钥,且不可能从加密密钥推导出解密密钥,加密密钥通常公开发布,而解密密钥则需要秘密保存。

一个公钥密码体制由 4 部分构成:明文(用 M 表示);密文(用 C 表示);公钥和私钥对,公钥用于加密,记为 K_e,此密钥公开,私钥用于解密,记为 K_d,此密钥保密,且从公钥很难推导出私钥;加、解密算法,算法公开,持有公钥的任何人都可以加密消息,只有持有私钥的人才能够解密。

一般情况下,网络中的用户预先约定一个共同使用的公钥密码系统,每个用户都有自己的一对公钥和私钥,网络中会有一个公开的数据库,任何用户都可以把自己的公钥发布到公开数据库中,同时可以从该数据库下载通信对方的公钥。基于公钥密码体制的一次秘密通信过程如图 2.14 所示,这里用户 Alice 利用公钥密码系统向用户 Bob 发送消息,具体步骤描述如下。

(1) Alice 从公开数据库中取出 Bob 的公钥 K_e;

(2) Alice 用 Bob 的公钥加密消息,并发送给 Bob:$C=E(M,K_e)$;

(3) Bob 用他的私钥 K_d 解密,获得 Alice 发送的消息:$M=D(C,K_d)$。

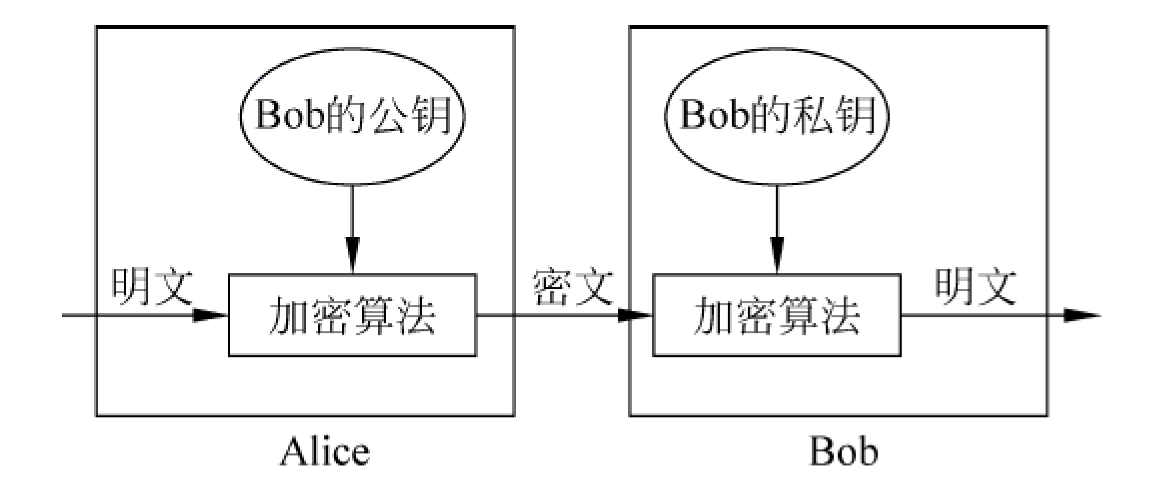


图 2.14　公钥密码体制下的秘密通信过程

2. 公钥密码体制的优点

相对于对称密码体制,公钥密码体制具有如下优点。

(1) 简化了密钥分配与管理。在对称密码体制中,加、解密密钥相同,用户两两通信必须采用不同的密钥,因此在采用该加密机制的通信网中,需要管理的密钥数量大,而采用公

钥密码体制进行保密通信，每个用户只需要一对公私钥，相对于对称密码体制，密钥的分配与管理得到了简化。

(2) 密钥不需传递。在对称密码体制中，由于加、解密密钥相同，因此在通信开始之前，需要安全传递密钥到接收方，受当前经济和技术条件的限制，密钥传递的绝对安全性无法得到保证。而在公钥密码体制中，存在一对密钥，加密密钥公开，解密密钥自己保存，不涉及密钥的保密传递，降低了密钥泄露的风险。

(3) 能提供不可否认性服务。对称密码体制无法防止通信双方的相互欺骗，而公钥密码体制不仅可以实现保密通信，而且还能实现提供不可否认性服务。在公钥密码体制中，用公钥加密的信息只能用对应的私钥解密，反之亦然，而私钥是用户私密保存的，因此如果某密文能用该用户的公钥解密，则可说明该消息一定是用该用户的私钥加密的，而私钥只有该用户知道，任何其他人都无法伪造，所以该消息一定来自该用户，无法抵赖。提供此功能的公钥密码算法称为数字签名，后续章节会详细介绍。

3. 公钥密码体制应该满足的要求

Differ 和 Hellman 同时在文中给出了公钥密码体制应该满足的要求：

(1) 用户产生一公私钥对(P_K,S_K)在计算上是容易的。

(2) 发送方利用接收方的公钥 P_K 对消息 M 进行加密产生密文 C，即 $C=E_{P_K}[M]$在计算上是容易的。

(3) 接收方利用自己的私钥 S_K 对密文 C 解密，即 $M=D_{S_K}[C]$，在计算上是容易的。

上面的三个条件是公钥密码体制的工程实用条件，因为只有算法高效，密码才能实际应用，否则，只有理论意义而无实用价值。

(4) 攻击者由公开密钥 P_K 求私钥 S_K 在计算上不可行。

这个条件是公钥密码的安全条件，是公钥密码的安全基础，也是最难满足的一个条件。

(5) 加、解密操作的次序可以互换，即 $E_{P_K}[D_{S_K}(M)]=D_{S_K}[E_{P_K}(M)]$。

自 1976 年公钥密码的思想被 Differ 和 Hellman 提出后，由于其优良的密码学特性和广阔的应用前景，吸引了全世界的密码爱好者，他们提出了各种各样的公钥密码算法和应用方案，密码学进入了一个空前繁荣的时代。然而研究公钥密码并非易事，尽管提出的方案有很多，但能经受得住时间考验的却寥寥无几。经过三十多年的研究和发展，目前世界公认的比较安全的公钥密码有 RSA 和 ElGamal 密码类。RSA 算法的安全性基于 100 位十进制数以上的大整数的素数因子分解难题，这是一个至今没有有效快速算法的数学难题。ElGamal 的安全性基于计算离散对数的困难性，离散对数问题是指模指数运算的逆问题，即找出一个数的离散对数，而计算离散对数是非常困难的。

2.5.3 RSA 公钥密码体制

RSA 是在 1977 年由美国麻省理工学院的三位科学家 Ron Rivest、Adi Shamir 和 Leonard Adleman 提出的非常著名的公钥密码算法，RSA 这个名字就来自他们的姓名缩写。

RSA 算法的安全性基于数论中将大整数分解成素数乘积的困难性，只要其密钥长度足够长，用 RSA 加密的信息实际上是不能被破解的。到目前为止，世界上还没有任何可靠的攻击 RSA 算法的方式。

RSA 密码既可用于加密，又可用于数字签名，RSA 密码已经成为目前应用最为广泛的公钥密码。许多国际标准化组织，如 ISO、ITU 等都已经将 RSA 作为标准。Internet 上的 E-mail 保密系统以及国际 VISA 和 MASTER 组织的电子商务协议 SET 都将 RSA 作为传送会话密钥和数字签名的标准。

1. RSA 算法

如果把加密算法看作函数、把明文看作定义域、把密文看作值域，那加密就是求函数 $F(X)$，解密实际上就是 $F(X)$的逆 $F^{-1}(X)$。

如果我们找到一个函数，求 $F(X)$容易，求 $F^{-1}(X)$难，这意味着只能加密、不能解密，这样的函数我们称为单向函数。简单来讲，单向函数就像是我们把蓝色的颜料和红色的颜料混合在一起很容易，但是混合完了之后再想分开就非常困难了。单向函数只能加密不能解密是没有意义的。如果这个单向函数在知道某些信息的情况下可以求出逆函数，那不就可以实现解密了吗？我们把这种信息称为陷门，知道它能够解密，不知道它无法解密，把存在陷门的函数称为单向陷门函数。所谓单向陷门函数就是在不知陷门信息下是单向函数，当知道陷门信息后，求逆是易于实现的。

如果我们找到这个单向陷门函数，发送方就可以利用这个单向陷门函数来做加密，但只有接收方才能利用不公开的陷门来解密从而实现加解密分离。这样我们的目标就转变为寻找一个单向陷门函数，也就是把一个密码学问题转换成了一个数学问题。

RSA 算法采用模函数 $M^k \bmod n$ 构造单向函数，在加密时，明文 M 经过加密运算得到密文 $C=M^e \bmod n$，e 为加密密钥。

密文经过解密得到明文 M：$C^d \bmod n=(M^e \bmod n)^d \bmod n=M^{ed} \bmod n=M$，$d$ 为解密密钥。

即必须存在 e、d、n，使 $M^{ed} \bmod n=M$ 成立。这里以 n、e 为公钥，私钥为 d。那么，现在的问题是，如何才能找到能够使 $M^{ed} \bmod n=M$ 成立的参数 e、d、n 呢？这就需要借助数论中的欧拉函数和欧拉定理。

欧拉函数：小于 n 且与 n 互素的正整数的个数，记为 $\Phi(n)$。如果 n 是素数则 $\Phi(n)=n-1$，若 $n=p\times q$（p、q 都是素数），则 $\Phi(n)=\Phi(p\times q)=\Phi(p)\times\Phi(q)=(p-1)\times(q-1)$。

欧拉定理：对于任意互素的整数 M 和 n，有 $M^{\Phi(n)}\equiv 1 \bmod n$，这里 $\Phi(n)$为欧拉函数。

根据欧拉定理，有 $M^{K\Phi(n)+1}\equiv M \bmod n$，这样，为了使 $M^{ed} \bmod n=M$ 成立，需要满足 $ed=K\Phi(n)+1$，也即 $ed \bmod \Phi(n)=1$，满足上式的 e、d 一定存在吗？根据数论中的乘法逆元定义：

如果 a，b 互为素数，存在 a^{-1}使得$(a\times a^{-1}) \bmod b=1$。

如果 e 和 $\Phi(n)$互素，则一定存在 d，使得 $ed \bmod \Phi(n)=1$ 成立。这样如果选取两大素数 p 和 q，$n=p\times q$，计算 n 的欧拉函数值 $\Phi(n)=(p-1)\times(q-1)$，随机选择一整数 e，使得 $1<e<\Phi(n)$和 $\gcd(\Phi(n),e)=1$ 成立，计算 e 模 $\Phi(n)$的乘法逆元，即为 d：$ed\equiv 1 \bmod \Phi(n)$，将 e 和 n 作为公钥公开，攻击者由已知的公钥获得私钥 d 的唯一途径是求得 $\Phi(n)$，而求$\Phi(n)$必须将 n 分解成两个素数的乘积，这是数论中的难题，没有有效的解决方法，由此保证了算法的安全性。

根据上述原理，RSA 算法流程可描述如下。

(1) 选两个保密的大素数 p 和 q（各为 100～200 位十进制数）。

(2) 计算 $n=p\times q$，$\Phi(n)=(p-1)\times(q-1)$，其中 $\Phi(n)$ 是 n 的欧拉函数值。

(3) 选一整数 e，满足 $1<e<\Phi(n)$ 和 $\gcd(\Phi(n),e)=1$ 成立。

(4) 计算 d，满足 $ed\equiv 1 \bmod \Phi(n)$。即 d 是 e 在模 $\Phi(n)$ 下的乘法逆元，因 e 与 $\Phi(n)$ 互素，由模运算可知，它的乘法逆元一定存在。

(5) 以 $\{e,n\}$ 为公钥，$\{d,n$ 为私钥$\}$。

【例 2-8】 已知 $p=7$，$q=17$，明文 $m=19$，求用 RSA 加密后的密文。

解：

求得 $n=p\times q=119$，$\Phi(n)=(p-1)\times(q-1)=96$

取 $e=5$，满足 $1<e<\Phi(n)$，且 $\gcd(\Phi(n),e)=1$。确定满足 $ed\equiv 1 \bmod 96$ 且小于 96 的 d，因为 $77\times5=385=4\times96+1$，所以 $d=77$。

因此公钥为 $\{5,119\}$，私钥为 $\{77,119\}$，明文为 $m=19$，则密文为：

$C=19^5 \bmod 119=66$

解密为 $66^{77} \bmod 119=19$

2. RSA 的安全性

RSA 算法的加密函数是一个单向函数，所以对于攻击者来说，试图解密密文在计算上是不可行的，因此对于 RSA 算法的攻击方法有如下几种。

(1) 穷举法：也就是尝试所有的私钥。抵御穷举攻击的方法是使用长的密钥。例如 RSA 目前建议的密钥长度为 2048 位，但是密钥长度的增加也增加了加密、解密的复杂性，运行速度也会受到较大影响。

(2) 利用数学分析来进行破解：RSA 算法及公钥 e、n 公开，用于解密的私钥 d 和公钥之间满足 $ed\equiv 1 \bmod \Phi(n)$，因此攻击者为了获得私钥 d，必须要能由 n 计算 $\Phi(n)$，而 $\Phi(n)=(p-1)\times(q-1)$，$n=p\times q$，所以数学攻击法实质上是试图把大整数分解为两个素数的乘积。虽然大合数因子分解十分困难，但随着科学技术的发展，人们对大合数因子分解的能力在不断提高，而且分解需要的成本在不断下降。继 1994 年 4 月 RSA-129 被破译，1996 年 4 月 RSA-130 也被破译。1999 年 2 月由美国、荷兰、英国、法国和澳大利亚的数学家和计算机专家，通过 Internet 网，历时 1 个月，成功分解了 140 位的大合数，破译了 RSA-140，同年 RSA-155 被破解。2002 年，RSA-158 也被成功因数分解；2005 年，RSA-200 被破解。因此我们今天要使用 RSA 密码，首先应当采用足够大的整数 n，普遍认为，n 至少应该 1024 位，最好 2048 位。估计在未来一段比较长的时期，密钥长度介于 1024～2048 位之间的 RSA 是安全的。

2.6 消息认证

为达到某种目的，攻击者会采取各种攻击方法对信息系统进行攻击，这些攻击方法分为两类：被动攻击和主动攻击。被动攻击以获取信息为目的，不对数据信息作任何篡改，破坏消息的机密性，前面介绍的加密技术可以预防被动攻击。而主动攻击通过冒充、重放、篡改等手段改变发送的消息，破坏了消息的完整性，例如当合法用户 A 和 B 在传递消息时，通信链上的恶意攻击者 C，可能通过盗取用户密码等方式假冒 A 的身份向 C 发送消息；可能截

获消息并将消息篡改后再发送给 B；也可能伪造一条消息发送给 B；或者是把消息 M 保存下来，在某个特定的时间再发送给用户 B，这种攻击形式称为消息的延迟；也可能在以后的时段中多次将截获的消息 M 发送给 B，这种攻击形式称为消息的重放，以上是主动攻击的主要方式，认证是发现和检测主动攻击的重要手段。

认证的目的有两个：第一，验证消息的发送者和接收者是合法的，不是冒充的，这称为实体认证或身份认证，具体技术包括口令、智能卡、指纹、视网膜等手段；第二是验证消息本身的完整性，称为消息认证，验证消息在传送或存储过程中是否被篡改、伪造、重放或延迟等。身份认证在后续章节中会介绍，本节主要介绍消息认证技术。

消息认证通过认证符来认证消息的完整性，认证符由消息的发送方通过认证函数产生，并传递给接收方，接收方通过验证认证符以鉴别收到消息的真实性。对一个消息认证系统而言，关键在于认证函数的选择，即如何根据需要传输的消息产生能够对该消息进行鉴别的认证符，认证函数主要有以下三类。

(1) 加密函数：对消息进行加密得到密文，用密文作为消息认证符。

(2) 消息认证码 MAC(Message Authentication Code)：消息认证码是基于消息和密钥的公开函数，它产生定长的值，将该值作为认证符。

(3) 散列函数(Hash Function)：散列函数能将任意长的消息映射成小的固定长度的消息，该消息用作认证符，映射过程不需密钥的参与。

2.6.1　消息加密认证

对消息进行加密不仅可以实现保密通信，也可以达到对消息进行认证的目的，我们主要讨论对称密码体制如何实现认证。

如图 2.15 所示，采用对称密码体制进行消息认证，用户 A、B 在通信之前，首先商定密钥 K，消息 M 经对称加密算法加密后传送给 B，例如发送方发送的明文为"端午节子时在南京火车站会合"，如果攻击者截获了传输的密文信息，由于不知道密钥，因此不知道如何更改密文中的信息才能使明文产生预期的改变，所以只能通过任意修改消息以破坏消息。接收方如何判断消息是否被篡改过呢？如果消息 M 是具有某种语法特征的文本，则 B 可通过分析解密后的消息是否具有合理的语法结构来判断消息是否完整，没有被篡改过的消息通常有正确的语言结构和一定的含义，而篡改后的消息几乎不再具有合理的语法结构，通过这一点可以判断消息是否被篡改过，例如如果收到的消息经解密后为"端午 · * iipp—会合"，则可判断该消息被篡改了。

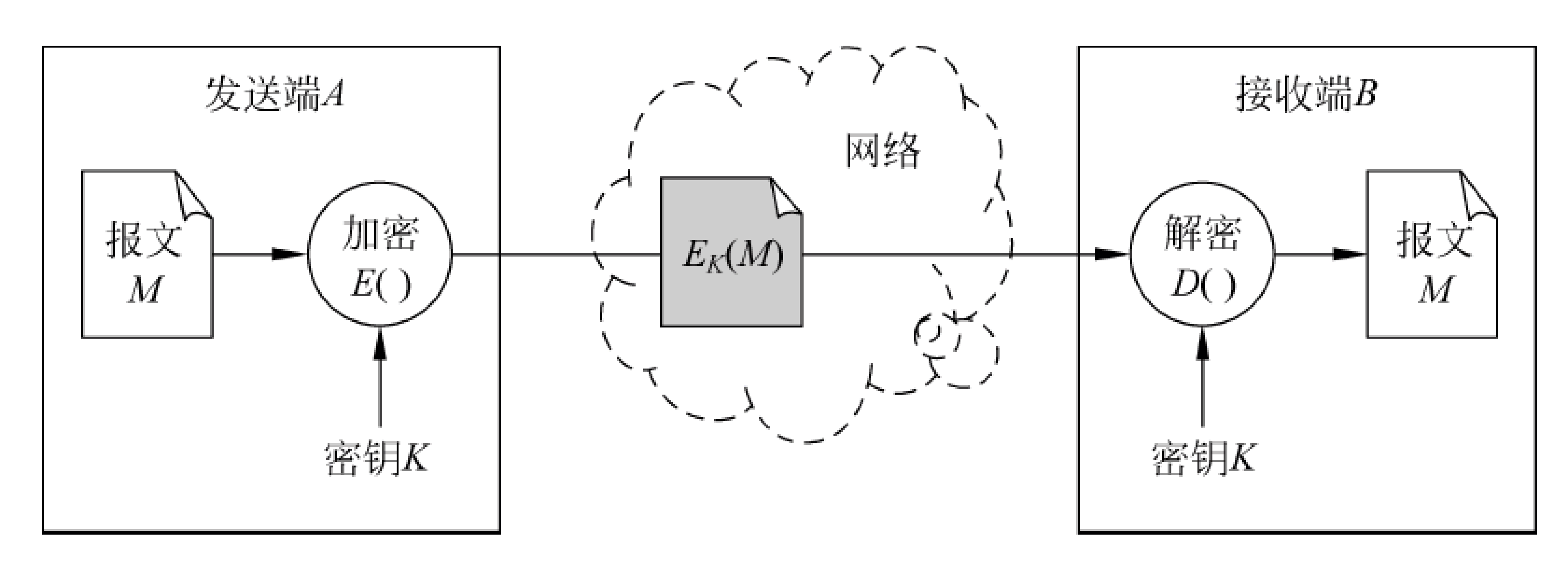


图 2.15　对称密码体制下的加密认证

如果消息 M 没有明显的语法结构或特征，例如二进制序列，采用上述方法则无法判断消息是否被篡改，为了解决这个问题，可强制使明文具有某种结构，例如可在消息上附加一个错误检测码 FCS，该错误检测码根据明文信息和公开的函数 F 产生，与消息串接后进行加密并传输，接收方在接收到信息后首先进行解密，分离出消息和 FCS，然后利用函数 F 对解密后的消息重新计算 FCS，如果跟传输过来的 FCS 相同的话，则说明消息没有被篡改，否则说明消息被篡改过，见图 2.16。

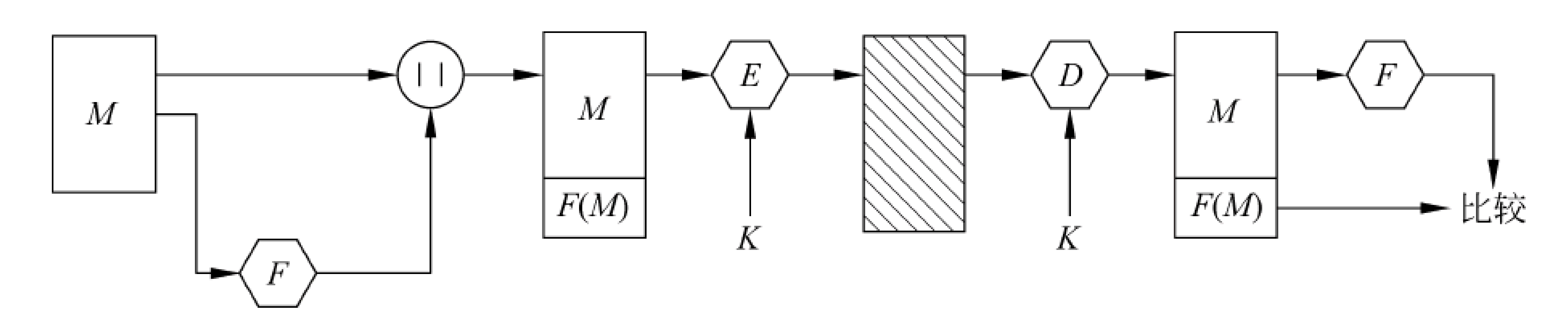


图 2.16 利用错误控制码的加密认证

采用加密函数实现认证，加密函数身兼数职，既要保证数据传输的保密性，又要认证报文的真实性。由于加密函数为了保证机密性通常设计复杂、计算代价大，而在有些情况下，只需要保证信息传输的真实性，而不需要机密性，例如政府或者权威部门的公告，如果将认证与加密分离则能够提供功能上的灵活性，下面介绍单纯提供真实性的认证机制——消息认证码和散列函数。

2.6.2 消息认证码

消息认证码是一种密钥相关的认证技术，它将密钥和消息一起代入认证函数，计算出一个固定长度的短数据块，称为 MAC，发送时将 MAC 附加在消息之后，一起发送给接收方。由于这种验证方式需要密钥的参与，而密钥只有通信双方知道，因此采用消息认证码不仅可以验证消息的完整性，而且还能对消息的发送方进行验证，具体认证过程如图 2.17 所示。

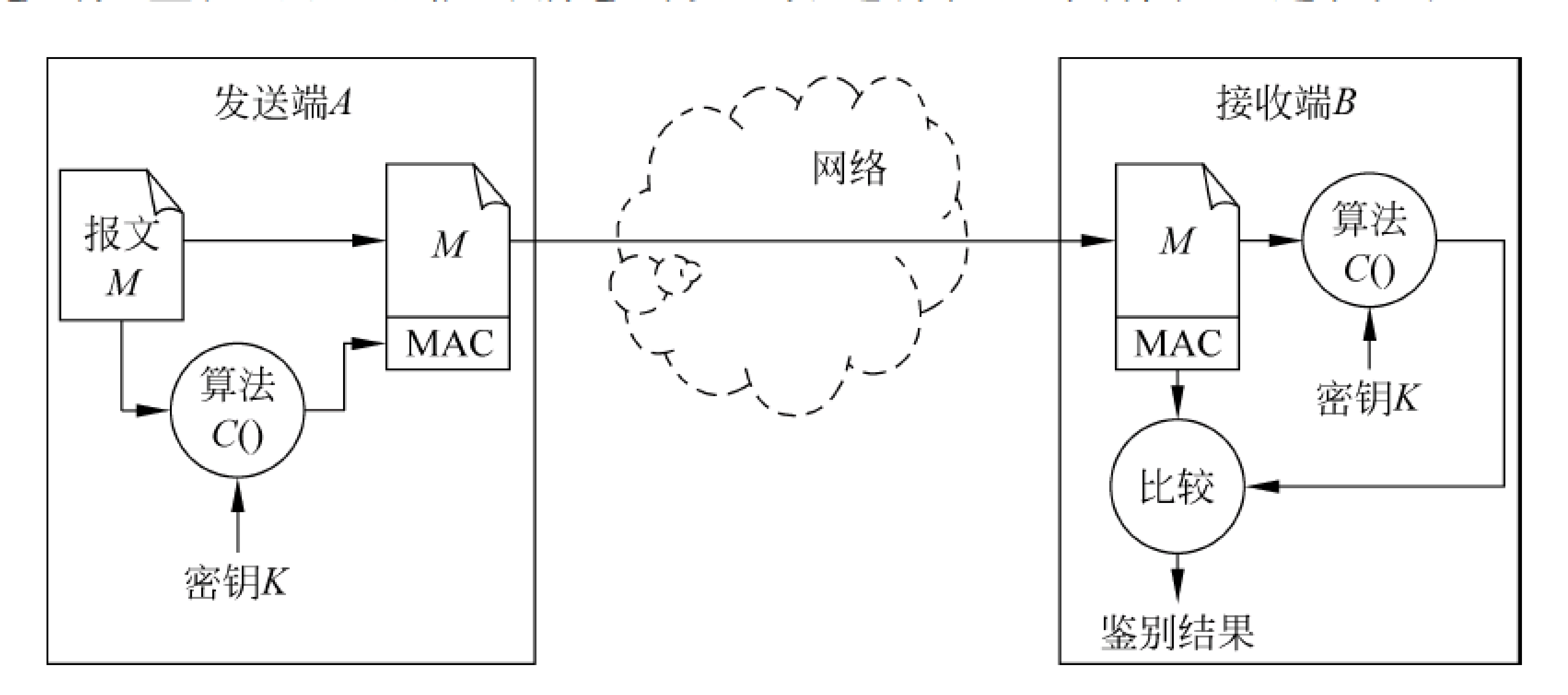


图 2.17 MAC 的基本用法

设 M 是发送方要发送的消息，K 是通信双方共享的密钥，则 $\text{MAC}=C_K(M)$，这里，C 是 MAC 函数，它将任意长的消息映射成短的定长的消息认证码 MAC。发送方将消息 M 和认证符 MAC 串接后一起发送给接收方。接收方在收到消息后，分离出消息 M 和 MAC，根据密钥、收到的消息 M 重新计算 MAC，并检查是否与传过来的 MAC 一致。如果两者相等，则接收者可以确信消息 M 未被篡改，因为如果攻击者改变了消息，由于不知道密钥 K，无法生成正确的

MAC,并且,如果接收者重新计算得到的 MAC 值与传送过来的 MAC 值一致的话,也可以确信消息来自真正的发送方,因为其他人由于没有密钥不能产生和原始消息相对应的 MAC。

MAC 函数与前面介绍的加密函数类似,在计算时都需要明文、密钥的参与,不同之处在于 MAC 算法不要求可逆性,而加密算法必须可逆,且相对于加密函数来说,MAC 的计算代价小,更适合进行消息完整性的认证。

在图 2.17 中,消息本身在传送时没有经过加密,不提供机密性,如果同时需要机密性,可以利用对称密码体制或者公钥密码体制对消息进行加密。

对消息认证码的攻击主要有两种方法:其一是攻击密钥,试图找到计算 MAC 的密钥;其二是攻击 MAC 函数的算法,找到其弱点。我们主要讨论攻击密钥的方法。

MAC 将任意长的消息映射为短的定长数据块,定义域空间大于值域空间,因此 MAC 函数是多对一的函数,即多个不同的消息可能映射到相同的 MAC 码,这将会增加密钥攻击的难度。如何找到 MAC 的密钥呢?假设攻击者已获得消息的明文和相应的 MAC,即已知(M_1,MAC_1),现要用穷举法来破解密钥,设密钥长度为 k,MAC 的长度是 n,通常 $k>n$,则所有可能的密钥个数是 2^k、所有可能的 MAC 个数为 2^n 个,因此大约有 2^{k-n} 个密钥对应相同的 MAC 码,这些密钥中哪个是正确的密钥呢,需要经过多轮测试才能找到正确的密钥,穷举攻击过程如下。

第一轮:

已知(M_1,MAC_1),for $i=1$ to 2^k,试探 $MAC_1=C_{Ki}(M_1)$,匹配数$\approx 2^{k-n}$,无法确定真正的密钥;

第二轮:

已知(M_2,MAC_2),for $i=1$ to 2^{k-n},试探 $MAC_2=C_{Ki}(M_2)$,匹配数$\approx 2^{k-2\times n}$,无法确定真正的密钥;

⋮

大约需要 k/n 轮才能找出一个唯一正确的密钥,所以用穷举法攻破 MAC 比攻破加密算法要困难得多。

2.6.3　Hash 函数

1. 安全散列函数的结构

Hash 函数又称为散列函数或杂凑函数,是一种能将不定长的输入映射成定长输出的特殊函数,记为 $h=H(M)$,其中 M 为消息,其长度可为任意,h 称为散列值、哈希值或消息摘要,长度一定,通常为 128 位或 160 位,生成散列值时不需要密钥。由于散列函数输入的长度是任意的,因此要求散列函数的计算效率比较高,即对于任何给定的消息 x,$H(x)$要相对易于计算。从 Hash 函数的特征,可以看出 H 是一个多对一的函数,多个不同的输入会产生相同的输出,因此散列函数必须具有单向性,也就是从 M 计算 h 容易,而从 h 计算 M 是不可能的。

对于两个差别很小(如仅差别一两位)的消息 Hash 函数产生的散列值会截然不同,因此 Hash 函数的一个重要功能就是实现消息完整性的检测,能够发现对消息的任何改动。例如,用户 A 和 B 通信,为了使得接收方能检测消息在传送过程中是否被篡改,用户 A 利用

散列函数对要发送的消息生成 Hash 值，附着在消息之后进行发送，接收方收到消息后，重新计算 Hash 值，如果跟发送过来的 Hash 值相同，则表示在传送过程中消息未被篡改，反之则表明消息被篡改，如图 2.18 所示。

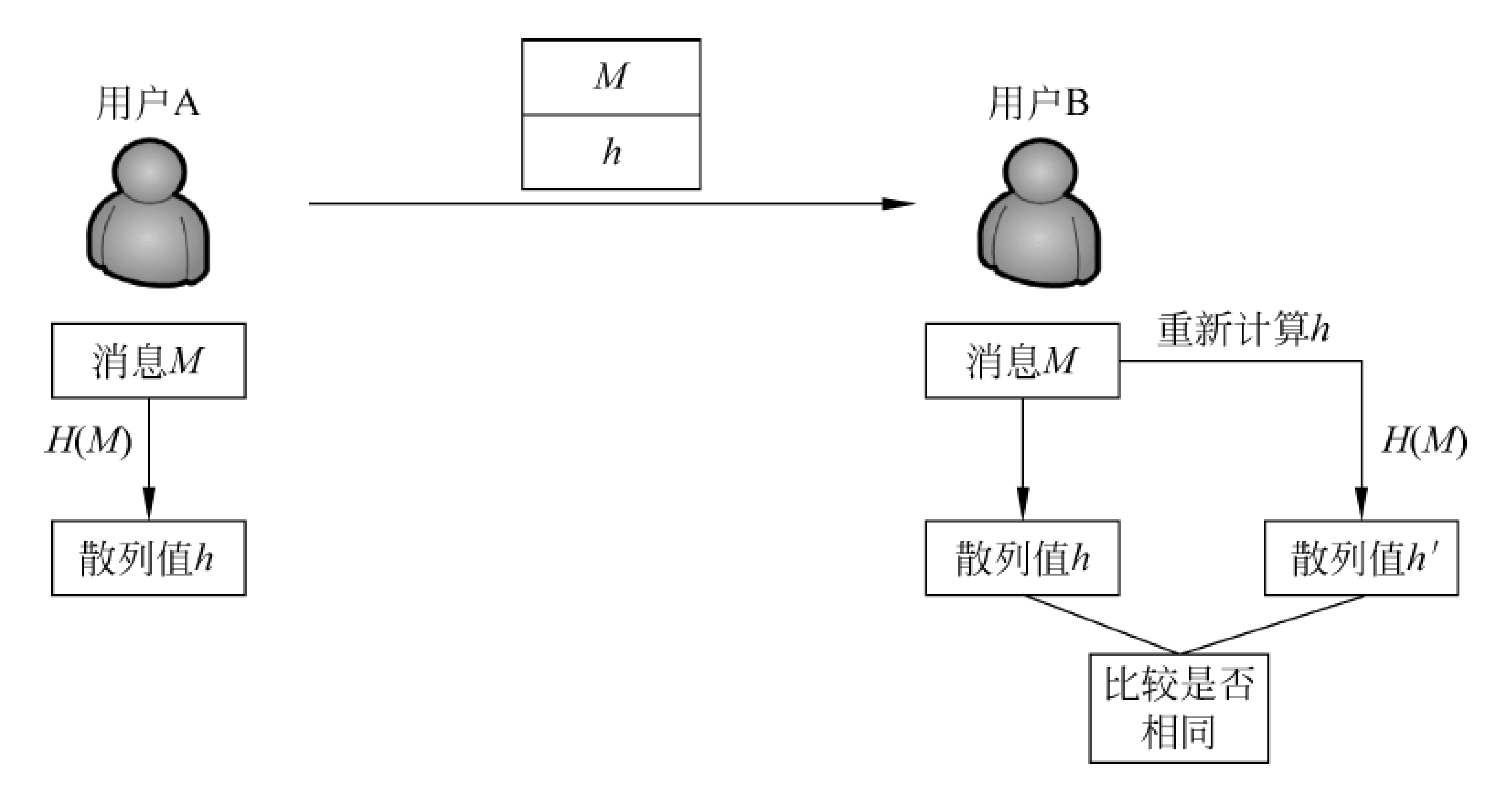


图 2.18　散列函数的简单用法

但是，由于散列函数计算时不需要密钥，并且通常散列函数属于公开函数，如果恶意用户在篡改明文 M 后，重新基于修改后的消息计算散列值，则接收方无法发现消息被篡改过，采用这种方法无法达到认证的效果。所以在实际使用过程中，发送方会用某种加密算法对 Hash 值进行加密，然后附着在消息后传递给接收方。接收方首先基于接收到的消息计算出 Hash 值，然后对传送过来的附加在消息后的 Hash 密文进行解密得到原始 Hash 值，如果两者相同则表示认证成功，如图 2.19 所示。

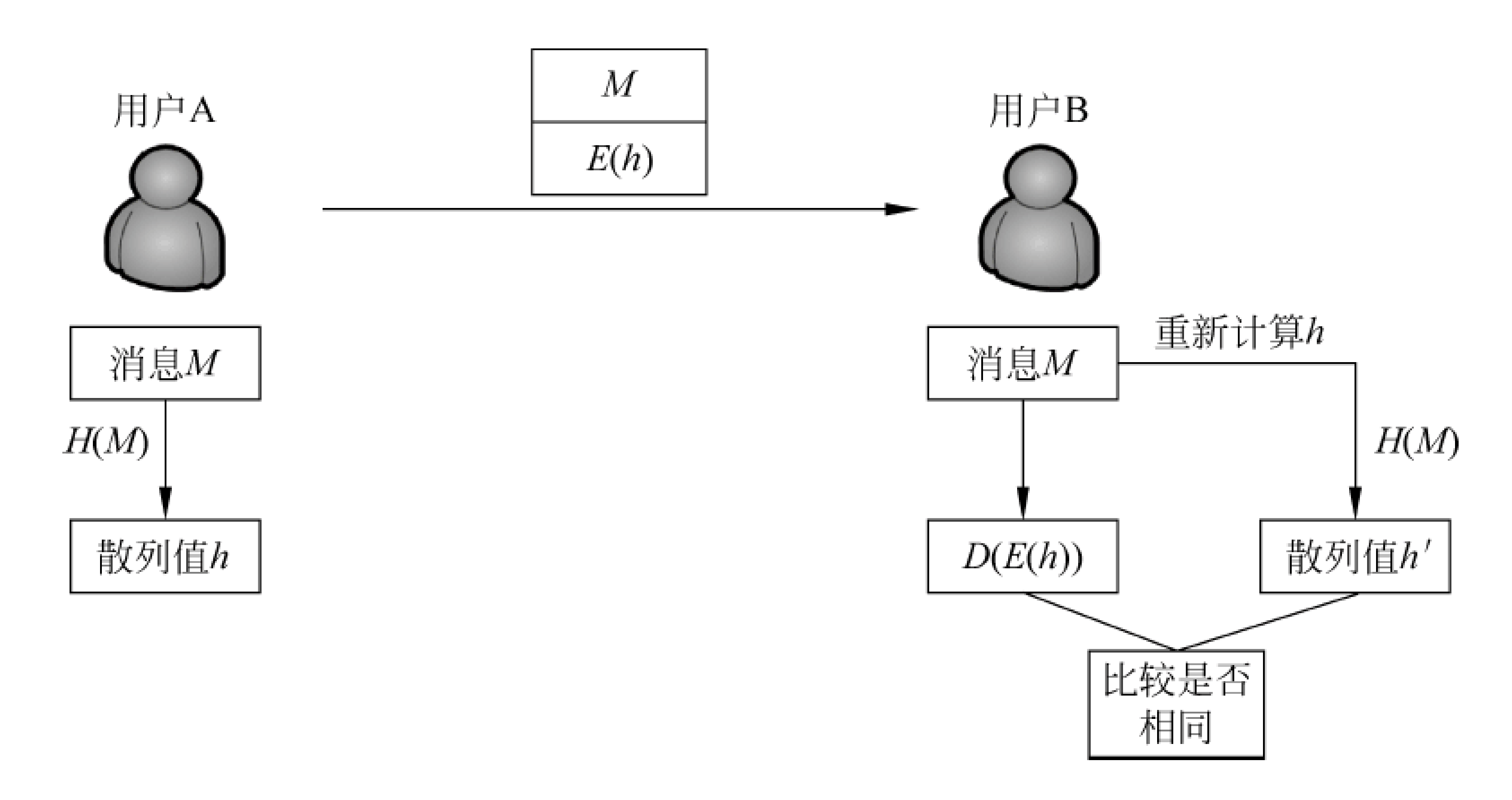


图 2.19　散列函数的改进用法

由于 Hash 值位数很少，一般只有 128 或者 160 位，所以加密或解密不会为通信系统带来太大负担。

为了防止第三方伪造 Hash 值或者通过 Hash 值计算出明文，散列函数 H 必须满足以下性质：

(1) 单向性：对任何给定的散列码 h，寻找 x 使得 $H(x)=h$ 在计算上不可行。

(2) 弱抗碰撞性(Weakly Collision-Free)：对于给定的 x，寻找不等于 x 的 y，使得

$H(x)=H(y)$在计算上是不可行的。

(3) 强抗碰撞性(Strongly Collision-Free):寻找任何的(x,y),使$H(x)=H(y)$在计算上是不可行的。

Hash 函数一般采用迭代的构造方法,其结构如图 2.20 所示,输入数据被划分为长度为 b 的分组,最后一个分组需要填充以满足长度要求,且最后一个分组包含了散列函数的输入总长度,散列算法中重复使用了一个压缩函数 f,f 的输入是前一轮的 n 位输出(称为链接变量)以及当前的 b 位分组,输出为 n 位的链接变量值。

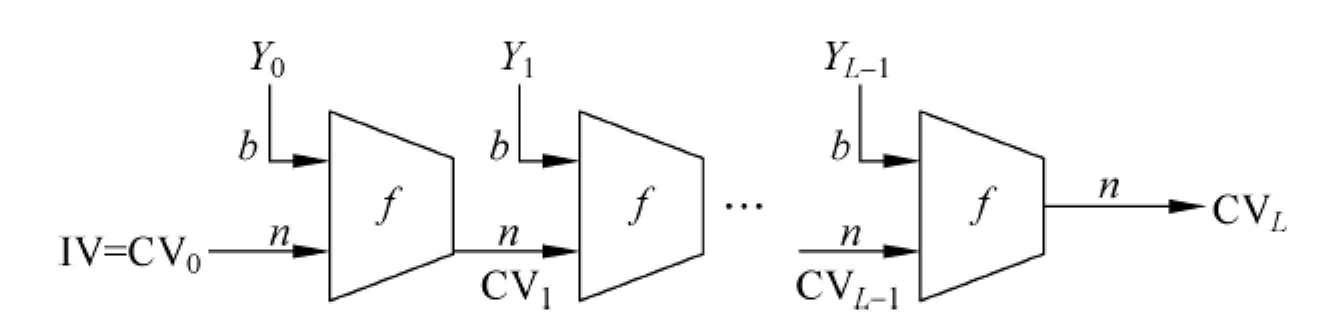


图 2.20 迭代型 Hash 函数的一般结构

图 2.20 中明文被分为 L 个分组 $Y_0, Y_1, \cdots, Y_{L-1}$,$b$ 为明文分组长度,n 为输出 Hash 值的长度,CV_i 是各级输出,IV 为 n 位的链接变量初始值,最后一个输出值即为 Hash 值。

迭代型结构的 Hash 函数已被证明是合理的,如果采用其他结构构造 Hash 函数不一定能确保安全性。采用这种结构的典型算法包括 MD5、SHA 等。MD5 算法由 RSA Data Security 公司的 Rivest 于 1992 年提出,能对任意长度的输入消息进行处理,产生 128b 长的消息摘要。SHA 算法是由美国国家标准技术研究所与国家安全局共同设计的安全哈希算法(Security Hash Algorithm,SHA),它能为任意长度的输入产生 160b 的散列值。这两个算法目前广泛应用于因特网的消息认证与数字签名中。这里主要介绍 MD5 算法。

2. MD5 算法

MD5 的全称是 Message-digest Algorithm 5(消息摘要算法),经 MD2、MD3 和 MD4 发展而来。利用 MD5 Hash 算法产生消息摘要时,对输入按 512b 的分组为单位进行处理,处理后输出为 128b 的 Hash 值。

如图 2.21 所示,算法处理过程主要包含以下几个步骤。

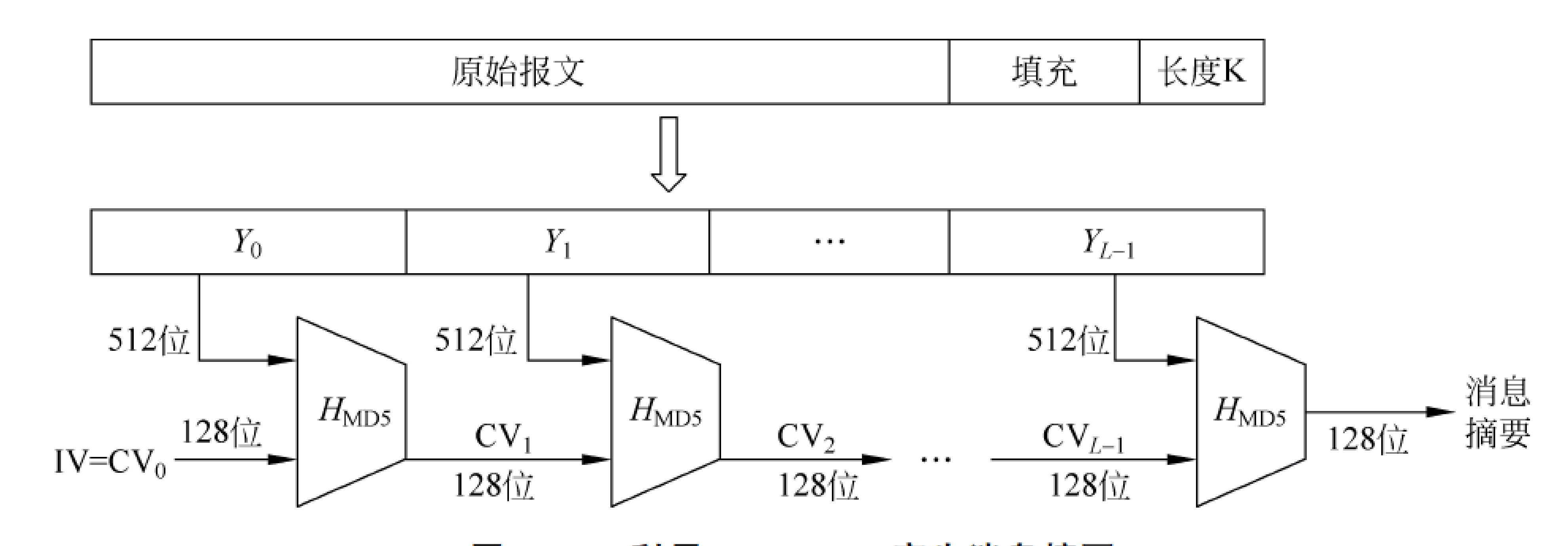


图 2.21 利用 MD5 Hash 产生消息摘要

1) 消息填充

首先填充消息使其长度比 512 的整数倍少 64 位。注意,即使消息本身已经满足上述长度要求,仍然需要进行填充。例如,若消息长度为 448 位,则仍需要填充 512 位,填充的内容由一个 1 和后续的多个 0 组成。

2）添加原始消息长度

在填充后的消息后面再添加一个64位的二进制整数表示填充前消息的长度。如果消息长度大于2^{64}，则取其对2^{64}的模。

执行这一步骤后，消息的长度为512的整数倍（设为L倍），则可将消息表示为分组长为512的一系列分组$Y_0,Y_1,\cdots,Y_{L-1}$。

3）缓冲区的初始化

Hash函数的中间结果和最终结果保存于128位的缓冲区中，缓冲区中的值称为链接变量，缓冲区由4个32位的寄存器组成，寄存器中的值分别用4个32b长的字表示A、B、C、D，其初始值分别为：

A=0x01234567；

B=0x89ABCDEF；

C=0xFEDCBA98；

D=0x76543210。

这些值以高端格式存储，即字节的最高有效位存于低地址字节位置。

4）H_{MD5}运算

压缩函数H_{MD5}对每个分组Y_q进行处理，是算法的核心，函数包括4轮处理过程，如图2.22所示。4轮处理过程结构一样，只是各轮所用的逻辑函数不同，各轮使用的逻辑函数依次记为F、G、H、I。每轮的输入为当前处理的消息分组Y_q和缓冲区当前值A、B、C、D，输出仍然放在缓冲区中以产生新的A、B、C、D，第四轮的输出和第一轮的输入相加得到最后的输出。

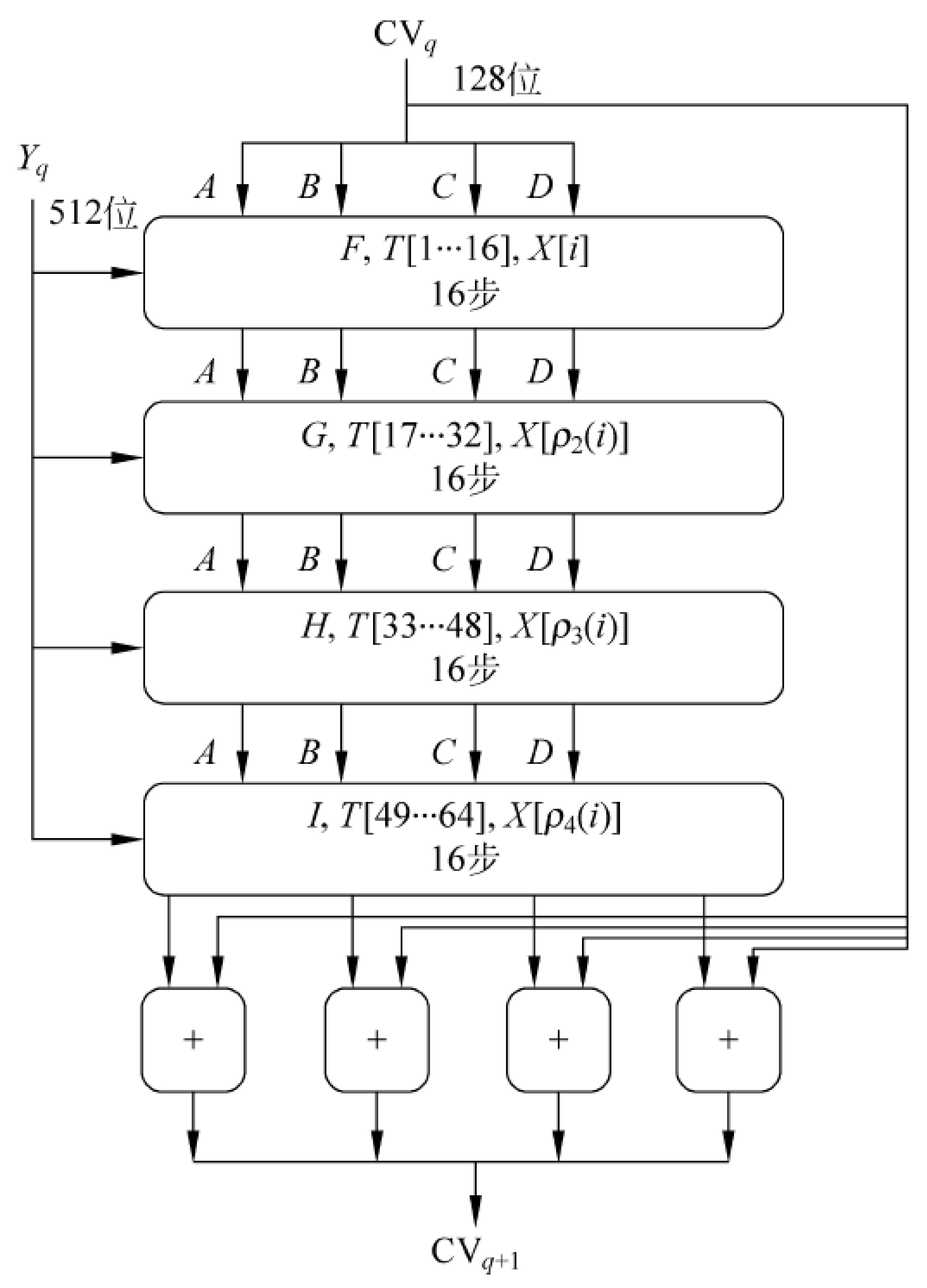


图2.22 一个分组的H_{MD5}处理

MD5 的每轮又要进行 16 轮迭代运算，4 轮共需 64 步完成，一步迭代过程如图 2.23 所示，首先当前缓冲区 B、C、D 中的链接变量执行非线性逻辑函数 $g(b,c,d)$，4 轮运算中使用的逻辑函数均不同，各轮逻辑函数的定义如表 2.19 所示。逻辑函数运算结果依次加上缓冲区 A 中的链接变量、消息的一个子分组和一个常数，再将所得的结果向左循环一个不定数，最后得到的结果再加上之前保存在缓冲区 B 中的值更新到缓冲区 B 中，原来缓冲区 D 中的值更新到缓冲区 A 中，原来缓冲区 B 中的值更新到缓冲区 C 中，原来缓冲区 C 中的值更新到缓冲区 D 中。每轮所使用的 512b 的消息分组 Y_q 被均分为 16 个子分量(每个子分组为 32b)，记为 $X[k]$，$k=0,1,\cdots,15$，表示当前分组的第 k 个 32 位字，在每一轮的运算中恰好被使用一次，在不同轮中其使用的顺序不相同。在第一轮中，其使用顺序为初始顺序，第二至第四轮中其使用顺序由表 2.20 所示的置换确定，表中 i 的取值为 0～15，对应每一轮的第一步骤到第十六步骤。图 2.23 中的 $T[i]$为 $2^{32}\times \text{abs}(\text{Sin}(i))$的整数部分，$i$ 是弧度，$T[1,\cdots,64]$个元素表，分 4 组参与不同轮的计算，其主要作用是消除输入数据的规律性。每轮各步循环左移位数不同，其中第一轮的 1～4 步分别循环左移 7、12、17、22 位，剩余 12 步则分别重复左移 7、12、17、22；第二轮分别循环左移 5、9、14、20 位；第三轮分别循环左移 4、11、16、23 位；第四轮分别循环左移 6、10、15、21 位。

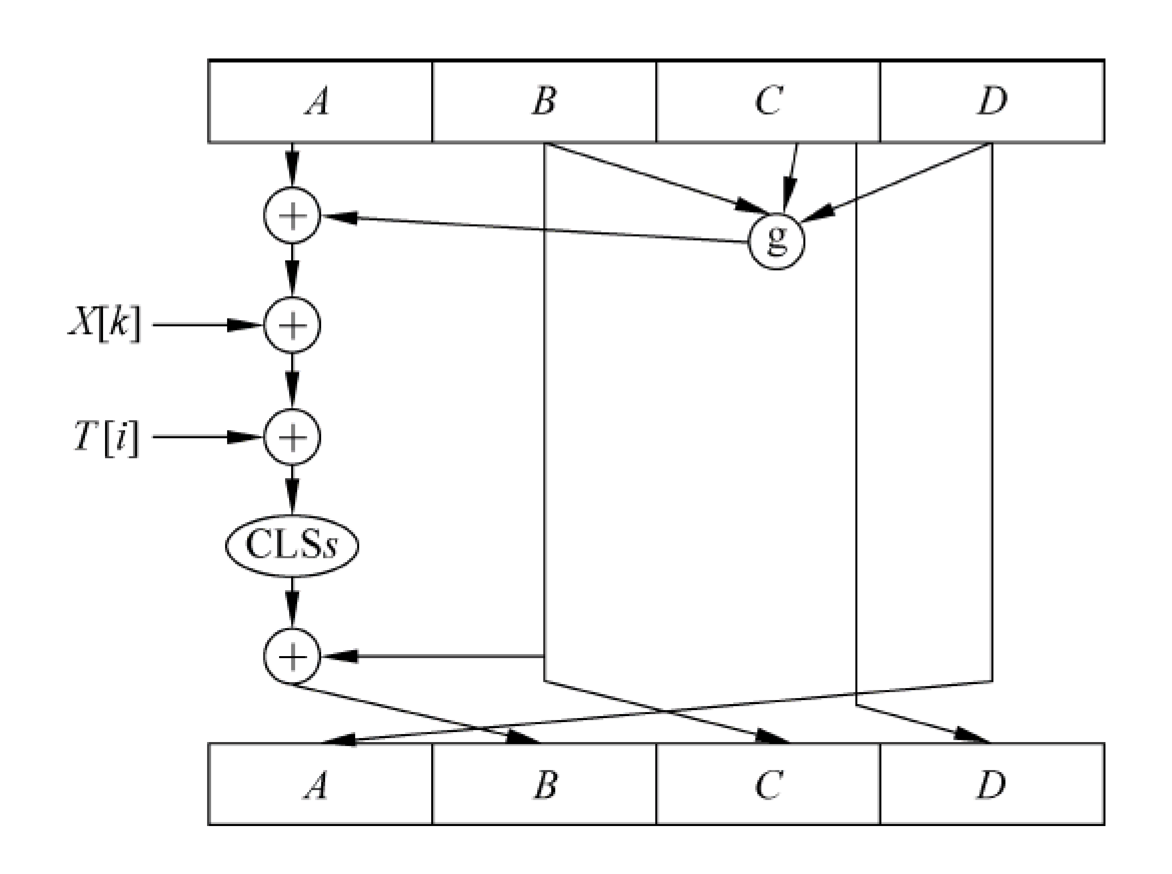


图 2.23　一步迭代

表 2.19　逻辑函数

轮	基本函数	$g(b,c,d)$
1	$F(b,c,d)$	$(b\wedge c)\vee(\bar{b}\wedge d)$
2	$G(b,c,d)$	$(b\wedge d)\vee(c\wedge \bar{d})$
3	$H(b,c,d)$	$b\oplus c\oplus d$
4	$I(b,c,d)$	$c\oplus b\vee \bar{d}$

表 2.20　明文分组使用顺序

明文子分组使用顺序	
轮数	计算公式
第二轮	$\rho_2(i)=(1+5i)\ \text{mod}\ 16$
第三轮	$\rho_3(i)=(5+3i)\ \text{mod}\ 16$
第四轮	$\rho_4(i)=7i\ \text{mod}\ 16$

通过这 64 步运算后，所得的结果与最初输入的分组进行模 2^{32} 加法，所得结果成为下一个分组进行运算的缓冲区初始值，以此类推。

所有 L 个 512 分组都处理完成后，最后一个分组的输出即为 128 位的消息。

3. 对 Hash 函数的攻击

一般来说，对一个 Hash 算法的攻击可分为三个级别。

(1) 预映射攻击(Preimage Attack)：给定 Hash 值 h，找到其对应的明文 M，使得 hash(M)＝h，这种攻击是最彻底的，如果一个 Hash 算法被人找出预映射，那这种算法是不能使用的。

(2) 弱碰撞攻击(Second Preimage Attack)：给定明文 M_1，找到另一明文 M_2($M_1 \neq M_2$)，使得 hash(M_1)＝hash(M_2)，这种攻击其实就是要寻找弱碰撞。

(3) 强碰撞攻击(Collision Attack)：找到 M_1 和 M_2，使得 hash(M_1)＝hash(M_2)，这种攻击其实就是要寻找强碰撞。

攻击者的主要目标是用非法消息替代合法消息进行伪造和欺骗，对哈希函数的攻击也是寻找碰撞的过程，要完成上述攻击行为，目前一般都是靠穷举的方法，因为那些没有通过分析和差分攻击考验的算法，大多都已经夭折在实验室了。下面分析采用穷举法进行弱碰撞攻击和强碰撞攻击的代价。

寻找弱碰撞的代价问题可以换种说法：给定一个散列函数 H 和某 Hash 值 h，假定 H 有 n 个可能的输出，如果 H 有 k 个随机输入，k 必须为多大才能使至少存在一个输入 y，使得 $H(y)=h$ 的概率大于 0.5?

因为 H 有 n 种可能的输出，所以对于某个 y 值，$H(y)=h$ 的概率为 $1/n$。那么 $H(y)\neq h$ 的概率为 $1-1/n$。那么随机地产生 k 个随机的 y 值，均使 $H(y)\neq h$ 的概率为 $(1-1/n)^k$，那么在 k 个随机的 y 值当中至少有一个使 $H(y)=h$ 的概率为 $1-(1-1/n)^k$。

根据二项式定理：$(1-a)^k=1-ka+k(k-1)a^2/2!\ -k(k-1)(k-2)a^3/3!\ +\cdots$

当 a 很小时，$(1-a)^k\approx 1-ka$，所以对于 $(1-1/n)^k$，当 n 很大时，$(1-1/n)^k\approx 1-k/n$。

那么在 k 个随机的 y 值当中至少有一个使 $H(y)=H(x)$ 的概率为 $1-(1-1/n)^k\approx k/n$。

现在要使这个概率等于 0.5，所以，$k=n/2$。

如果 Hash 函数为 m 位，则有 2^m 个可能的 Hash 码输出，如果给定 $h=H(x)$，要想找到一个 y，使 $H(y)=h$ 的概率为 0.5，尝试的次数大约为 $2^m/2=2^{m-1}$。

对于一个使用 64 位的 Hash 码，攻击者要想找到满足 $H(M')=H(M)$ 的 M' 来替代 M，即寻找一个弱碰撞，平均来讲，找到这样的消息大约需要进行 2^{63} 次尝试。这个结果似乎表明选择 64 位的散列函数是安全的，但事实并非如此。Yuval 提出的“生日悖论”能更有效地找到碰撞。

首先了解一下“生日悖论”的数学背景：k 为多大时，在 k 个人中至少有两个人具有相同生日的概率不小于 0.5?

分析：第一个人的生日占了一天，因此第二个人有不同生日的概率为 364/365，第三个人则少了两个选择，因此他与前两个人生日都不同的概率是 363/365，以此类推，第 k 个人与前面 $k-1$ 个人生日都不同的概率是 $[365-(k-1)]/365$。

所以，这 k 个人生日都不同的概率是 $(364/365)\times(363/365)\times\cdots\times((365-k+1)/365)$。

那么，这 k 个人中至少有两个生日相同的概率则为：

$$P = 1-(364/365)\times(363/365)\times\cdots\times((365-k+1)/365)$$
$$= 1-365!/(365-k)!(365)^k$$

可以计算出，当 $k=23$ 时，$p=0.5073$。

结果说明：任找 23 个人，从中总能选出两个人具有相同生日的概率至少为 0.5。这样的结果与人的直觉是违背的，这说明某些事情的发生概率是比我们的感觉要大得多的，这就是“生日悖论”。建立在“生日悖论”基础上的生日攻击，能够有效地找到强碰撞。

可以把找强碰撞的问题表述成：对于一个 64 位的 Hash 码，攻击者以 50% 的概率找到 M_1、M_2，使得 $Hash(M_1)=Hash(M_2)$，他要找到这样的消息大约要进行多少次尝试。

通过计算，大约要进行 $2^{64/2}$ 次尝试就能找到散列值相同的两个消息。这一计算代价要比寻找一个弱碰撞小很多。

如何实施生日攻击呢？假设 Hash 算法生成 64 位 Hash 值，攻击者可以采用如下方法来进行生日攻击。

攻击者截获到报文 M，根据 M 生成 2^{32} 表达相同含义的报文集 M_1（例如在文字中加入空格、换行字符等）。同时，攻击者还准备了 2^{32} 个用于欺骗的假报文 M_2。计算两个报文集中能够产生相同签名的报文对 $<M_1', M_2'>$，根据生日悖论，只要尝试次数 $k>2^{m/2}$ 成功的概率大于 0.5。攻击者将 M_1' 提交给发方请求签名，并用 M_2' 替代 M_1'。因为这两个报文具有相同签名，因此即使不知道加密密钥，攻击者也能够获得成功。

生日攻击表明 Hash 值的长度必须达到一定的值，如果过短，则容易遭受穷举攻击，一般建议 Hash 值需要 160 位，SHA-1 的最初选择是 128 位，后改为 160 位，就是为了防止利用生日攻击穷举 Hash 值。

2.7　数字签名

2.7.1　数字签名的定义

采用加密技术对传输的信息加密可以防止攻击者窃取信息；采用消息认证技术，消息接收方能验证消息内容是否被篡改过，用于保护通信双方的数据交换不被第三方侵犯，但是这两种技术都不能保证通信双方自身的相互欺骗，例如发送方 A 可以否认发送过消息，而接收方 B 也可以伪造一个不同的消息，但声称是从 A 收到的，这些行为称为信息抵赖。

签名具有防止抵赖的功能，它是证明当事者身份和数据真实性的一种信息。传统的军事、政治、外交活动中的文件、命令和条约及商业中的契约等采用书面签名的方式，如手印、签字、印章等，以表示确认和防止抵赖，书面签名得到司法部门的支持和承认，具有一定的法律效力。随着计算机通信网的发展，人们更希望通过电子设备实现快速、远距离交易，在以计算机文件为基础的现代事务处理中，应该采用电子形式的签名，即数字签名（Digital Signature），它是一种防止源点或终点抵赖的鉴别技术，用于防范通信双方的欺骗。数字签名在 ISO 7489-2 标准中定义为：附加在数据单元上的一些数据，或是对数据单元所做的密码变换，这种数据和变换允许数据单元的接收者确认数据单元的来源和数据单元的完整性，并保护数据，防止被人（例如接收者）伪造。美国电子签名标准（DSS，FIPS186-2）对数字签名作了如下解释：利用一套规则和一个参数对数据计算所得的结果，用此结果能够确认签

名者的身份和数据的完整性。数字签名通常利用公钥密码体制进行,其安全性取决于密码体制的安全程度。

在中国,数字签名是具有法律效力的,正在被普遍使用。2000 年,中华人民共和国的新《合同法》首次确认了电子合同、电子签名的法律效应。2005 年 4 月 1 日起,中国首部《电子签名法》正式实施。

2.7.2 数字签名的原理

在传统文件中,书面签名长期以来被用作用户身份的证明,表明签名者同意文件的内容。实际上,签名体现了以下几个方面的保证。

(1) 签名是不可伪造的。签名证明是签字者而不是其他人在文件上签字。

(2) 签名是不可重用的。签名是文件的一部分,不可能将签名移动到不同的文件上。

(3) 签名后的文件是不可改变的。在文件签名后,文件就不能改变。

(4) 签名是不可抵赖的。签名和文件是不可分离的,签名者事后不能声称他没有签过文件。

手印、签名、印章等传统的书面签名基本上满足以上条件,所以得到司法部门的支持。例如人的指纹具有非常稳定的特性,终身不变,据专家计算,大约 50 亿人才会有一例相同;公安部门有专业的机构进行笔迹鉴别;公章的刻制和使用都受到法律的保护和限制。

直接将书面签名扫描到计算机中在需要签名的地方将其粘贴上去,这种方法可行吗?这种方法实际是存在问题的。首先,将扫描的签名从一个文件剪辑和粘贴到另一个文件是很容易的,其次,文件在签名后也易于修改,并且不会留下修改的痕迹。所以,通过简单扫描书面签名作为数字签名不能满足上面提出的签名应该具有的条件,这种方法不可行。为了方便使用和实现,对数字签名提出的更进一步的要求如下。

(1) 依赖性:签名必须是依赖签名信息产生的。

(2) 唯一性:签名必须使用某些对发送者来说是唯一的信息,以防止双方的伪造和否认。

(3) 可验性:必须相对容易识别和验证该数字签名。

(4) 抗伪造:伪造该数字签名在计算上是不可行的,根据一个已有的数字签名来构造消息是不可行的,对一个给定消息伪造数字签名是不可行的。

因为数字签名具有不可伪造性,因此必须要用用户独有的信息产生,私钥是用户独有的信息,可以考虑通过私钥来产生数字签名,因此数字签名大多是基于公钥密码体制实现的。用公钥密码体制实现保密性安全目标的时候,公钥用于加密而私钥用于解密,而在进行数字签名的时候,应该用私钥产生签名信息而公钥用于验证用户的签名。具体的签名和验证方式如图 2.24 所示。采用公钥密码体制进行数字签名,发送者用自己的私钥加密数据后发送给接收者,接收者如果能用发送者的公钥解密数据,就可以确定消息一定来自于发送者,发送者对所发的信息不能抵赖。

发送方 A 利用 A 的私钥对整个发送的消息进行加密,生成的密文作为签名附加到消息之后发送给接收方 B。接收方 B 接收到消息后,分离出签名信息,利用 A 的公钥解密,解密后所得到的消息如果跟发送过来的消息一致就说明,该消息是用户 A 发送过来的,原因是该消息是用 A 的公钥解密得到的,只有用 A 的私钥加密的信息才能用 A 的公钥解密,而 A 的私钥信息只有 A 才拥有,所以该消息必定是由 A 发出来的,A 不能否认发送过该消息。

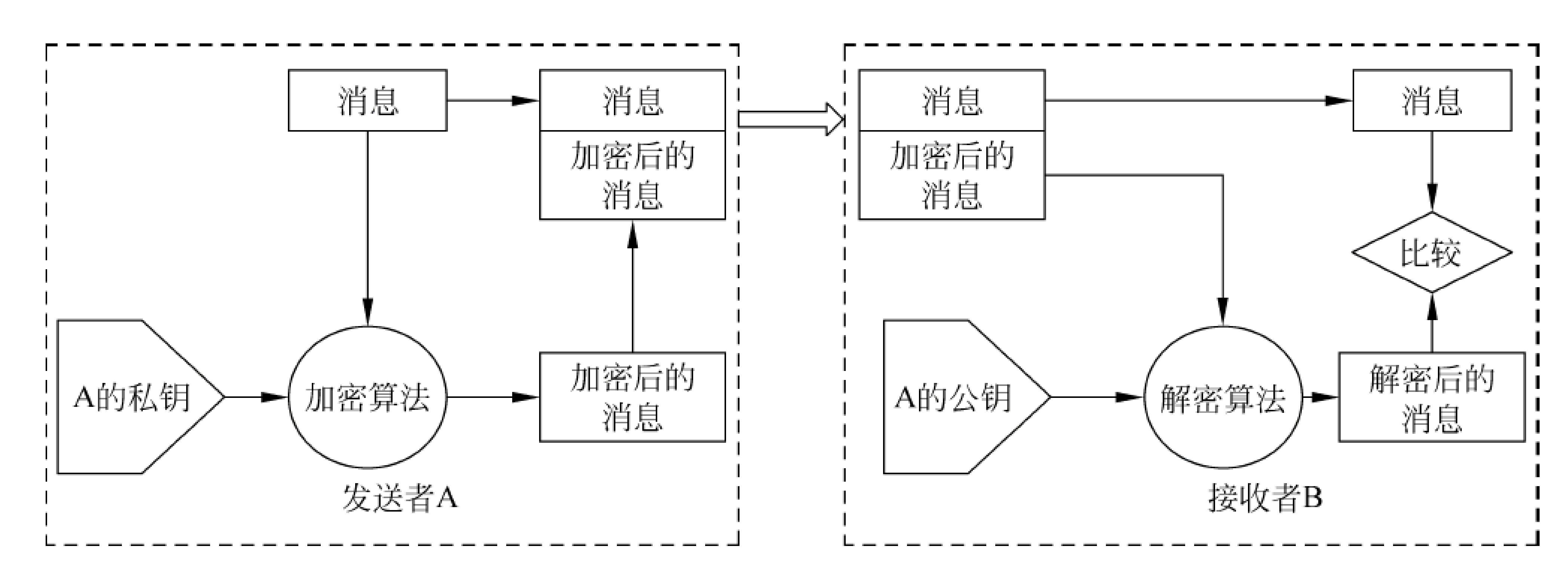


图 2.24　基于公钥密码体制的直接数字签名

基于公钥密码体制的直接数字签名的问题主要在于公钥密码体制为安全性考虑，本身参与计算的都是大数，因此执行效率并不高，怎样提高基于公钥密码体制的数字签名的效率呢？可以考虑减少加密的对象，由此联想到散列函数，散列函数可以将不定长的输入产生定长的输出，而且通常输出相对消息本身会小很多，因此可以考虑首先对消息通过散列函数得到消息的摘要，对消息摘要通过私钥进行签名，这就产生了基于公钥密码体制和散列函数的数字签名，签名过程如图 2.25 所示。

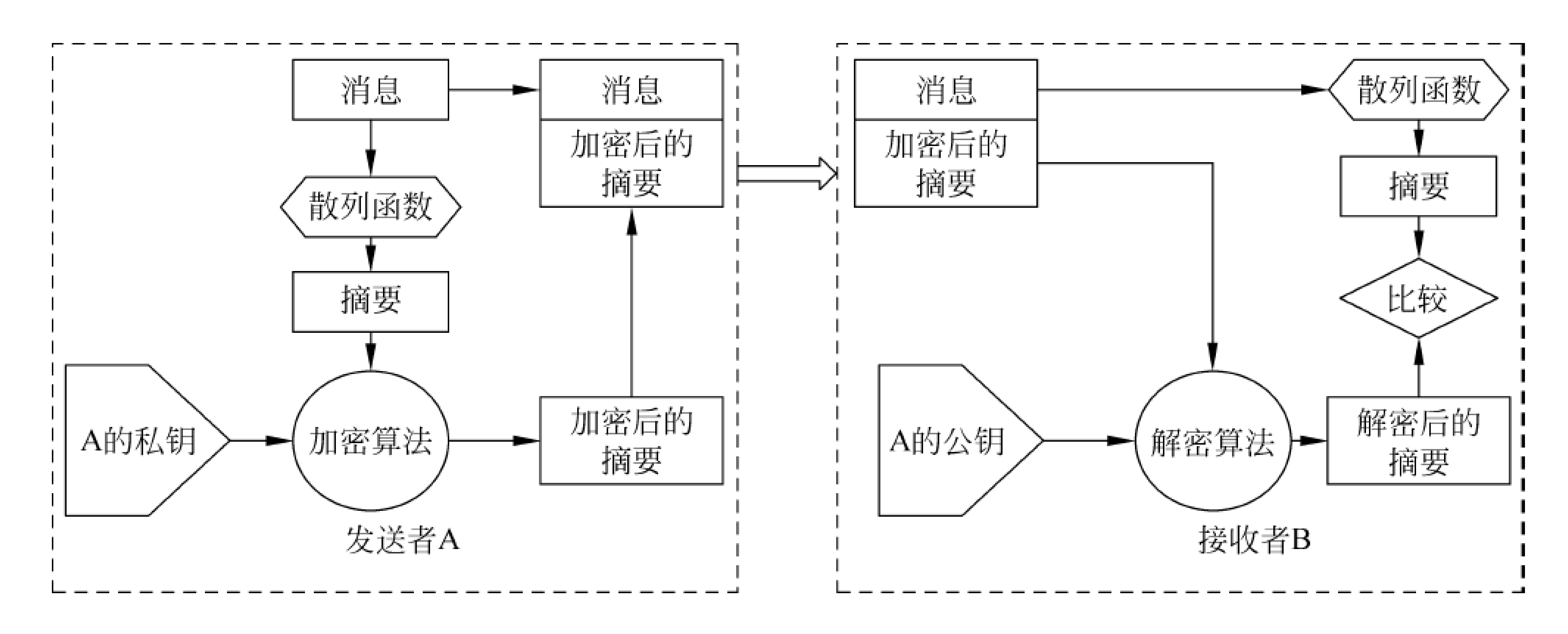


图 2.25　基于公钥密码体制和散列函数的直接数字签名

发送方 A 首先产生消息的摘要，利用 A 的私钥进行加密产生签名，将签名附加到消息之后，发送给接收方 B。接收方 B 在接收到信息后，分离出签名信息，用 A 的公钥解密得到消息摘要，接收方再基于接收的消息产生消息摘要，两者进行比较，如果一致，说明消息是发送方 A 发送的，并且 A 不可否认。

对于长度为 160b 的散列函数，两个不同的文件具有相同的散列值的概率为 $1/2^{160}$，所以在这个协议中使用散列函数的签名与使用文件的签名是一样安全的。

上述数字签名假定接收方知道发送方的公钥，签名通过使用发送方的私钥加密产生，整个签名和验证过程只牵涉到通信双方，属于直接数字签名，这种体制有个共同的弱点：方案的有效性依赖于发送方私钥的安全性。如果发送方随后想否认发送过某个数字签名消息，他可以声称用来签名的私钥丢失或者被盗用，并有人伪造了他的签名，通常需要采用与私钥安全性相关的行政管理控制手段来制止这种情况，但威胁依然存在。

为了解决直接数字签名中存在的问题，引入了仲裁数字签名。有关仲裁数字签名本文不作详细介绍。

2.7.3　数字签名的算法

数字签名的算法很多，应用最广泛的三种是RSA签名、DSS签名和基于ECC密码体制的ECDSA签名，这里对DSS签名稍作介绍。

DSS最初提出于1991年，1993年根据公众对于其安全性的反馈意见进行了修改，1994年美国国家标准技术研究所颁布了联邦信息处理标准FIPS 186，称为数字签名标准(DSS)，1996年又稍作修改，2000年发布了该标准的扩充版，即FIPS 186-2。

与RSA不同的是，DSS只用于数字签名，而不用于加密或密钥分发。DSS的数字签名算法是DSA，其安全性建立在求离散对数的困难性上，签名时首先利用安全散列函数SHA对消息M计算出散列值，对该散列值进行签名，签名时需要三类参数：为此次签名产生的随机数k、发送方的私钥SK_A和全局公钥PK_G，全局公钥PK_G为一组通信伙伴所共有，最终生成的签名由两部分组成，标记为s和r。

接收方在验证时首先基于接收到的消息产生散列值，将这个散列值和接收到的签名(s,r)一起作为验证函数的输入，验证函数依赖于全局公钥PK_G和发送方公钥PK_A，若验证函数的输出等于签名中的r，则签名是有效的。签名函数保证只有拥有私钥的发送方才能产生有效签名，如图2.26所示。

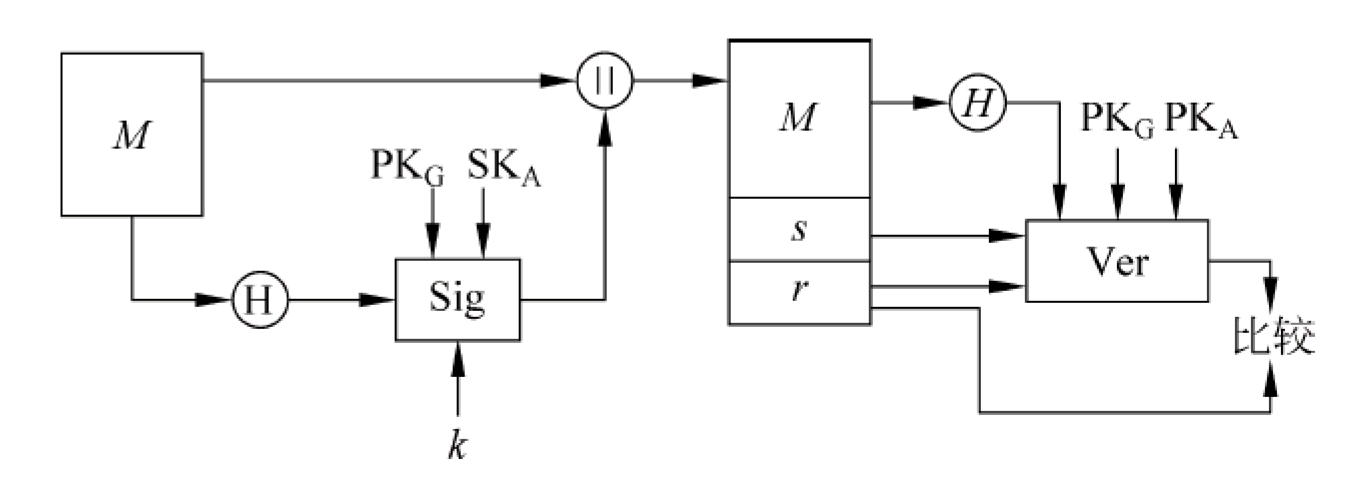


图2.26　DSS签名方案

有关于DSA算法的具体签名过程请参考相关书籍。

2.8　公钥基础设施PKI

2.8.1　公钥的分配

公钥密码体制有公、私两个密钥，在进行密钥分配时要确保私钥的秘密性、真实性和完整性，对于公钥，则保证其真实性和完整性，绝不允许攻击者替换或者篡改用户的公钥。

公钥的分配方式主要有以下几种。

(1) 公开发布：用户将自己的公钥通过BBS或者邮件列表等方式公开发布。这种方法方便快捷，每个人都可以很方便地发布自己的公钥，但容易被人冒充或者篡改，所以这种方法一般在简单的个人应用或者一些小型网络中使用。

(2) 公钥动态目录表：建立一个公用的公钥动态目录表，表的建立和维护以及公钥的

发布由某个公钥管理机构承担,每个用户都可靠地知道管理机构的公钥。但在这个方法中,每一用户想要与他人通信都要求助于公钥管理机构,因而可能形成瓶颈,而且公钥目录表也容易被篡改。所以这个方法只适合于小型网络,例如企业局域网中。

(3) 直接发送。通信双方直接将自己的公钥发送给对方,由于公钥本身就是公开的,因此发送时不需要加密处理。但是这种方式容易受到“中间人”(Man-in-the-Middle)攻击。例如,考虑 A 和 B 进行通信的情况。假设攻击者 C 能够截获公钥的交换,C 可以向 A 发送他自己的公钥,但故意将其表示成 B 的公钥,然后,他还可以向 B 发送自己的公钥,故意表示成 A 的公钥,现在 C 可以拦截 A 和 B 之间的所有通信了。如果 A 向 B 发送了一条加密消息,由于加密实际上使用的是 C 的公钥,C 获得消息后解密并存储,之后,他使用 B 的公钥加密其篡改后的消息,继续将其发送给 B,B 获得消息后能够解密,但不知道消息来自于 C 而非 A。

出现上述问题的实质是 A 没办法确定他得到的密钥是否真的属于 B,为了解决这个问题,需要借助于可信的第三方,通过颁发数字证书,将用户公钥和真实身份绑定。

(4) 数字证书:采用数字证书分配公钥是最安全有效的方法,数字证书由证书管理机构 CA 为用户建立,实际上是一个数据结构,其中的数据项有该用户的公钥、用户的身份和时间戳等。

2.8.2　数字证书

公钥是公开的,因此不需要确保秘密性,但是需要保证完整性和真实性,绝对不允许攻击者更换或篡改用户的公钥。如果公钥的真实性和完整性受到危害,则基于公钥的各种应用的安全将受到危害。保证完整性和真实性的方法之一就是数字签名。假设有一个权威公正的第三方可信任实体 X,所有的公钥都交由实体 X 验证签名后存入某个数据库公开发布,实体 X 通过可信途径把自己的公钥公开,则用户从数据库中取出公钥时通过验证实体 X 的签名是否完整,从而可以发现对公钥的篡改。进一步,如果将用户的标识符和用户的公钥联系在一起签名,则可以确认公钥的身份,防止有人冒充或者伪造公钥。

我们把可信的实体 X 称为签证机构(Certificate Authority,CA),由一个可信任的权威机构签署的信息集合称为证书,它实质是一个数据结构。证书分很多种类型,例如 X.509 公钥证书、简单的 PKI 证书、PGP 证书、属性证书等,这些证书具有各自不同的格式。

一个简单的数字证书示意如图 2.27 所示。

公钥证书包括持证主体的身份标识、公钥等相关信息,这些信息由签证机构进行数字签名,数字签名也是证书的一部分,任何知道签证机构公钥的人通过验证签名的真伪,确保公钥的真实性、确保公钥与持证主体之间的严格绑定。

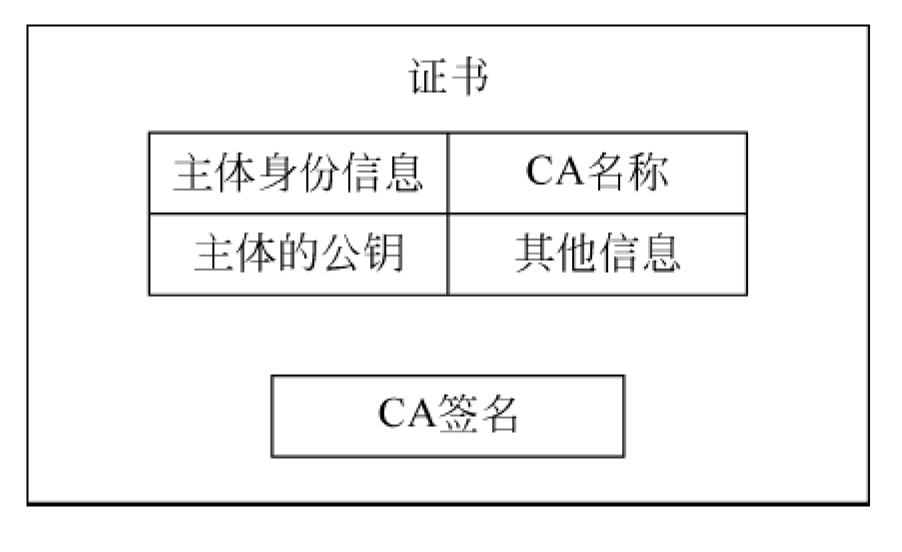


图 2.27　数字证书示意图

日常生活中有许多使用证书的例子,例如汽车的驾驶证。驾驶证(公钥证书)确认了驾驶员的身份(用户),表示其开车的能力(公钥),驾驶证上有公安局的印章(CA 对证书的签名),任何人只要信任公安局(CA),就可以信任驾驶证(公钥证书)。

有了公钥证书系统后,如果某个用户需要任何

其他已向CA注册的用户的公钥,可以向持证人(或证书机构)直接索取其公钥证书,并用CA的公钥验证CA的签名,从而获取可信的公钥。公钥证书为公钥的分发奠定了基础,成为公钥密码在大型网络系统中应用的关键技术。电子商务、电子政务等大型网络应用系统都采用了公钥证书技术。

2.8.3 X.509证书

目前应用最广泛的证书格式是国际电信联盟(Internet Telecommunication Union,ITU)提出的X.509版本3格式。X.509是由ITU制定的数字证书标准。为了提供公用网络用户目录信息服务,ITU于1988年制定了X.500系列标准。其中X.500和X.509是安全认证系统的核心,X.500定义了一种区别命名规则,以命名树来确保用户名称的唯一性,X.509则为X.500用户名称提供了通信实体鉴别机制,并规定了实体鉴别过程中广泛适用的证书语法和数据接口,X.509称为证书。

最初的X.509版本公布于1988年,版本3的建议稿1994年公布,在1995年获得批准。本质上,X.509证书由用户公钥与用户标识符组成,此外还包括版本号、证书序列号、CA标识符、签发者名称、证书有效期等。

X. 509版本3的证书结构如图2.28所示。

X.509证书
版本号
证书序列号
签名算法标识符
颁发者名称
有效期
主体名称
主体公钥信息(算法标识、公钥值)
颁发者唯一标识符(可选)
主体唯一标识符(可选)
扩展项(可选)
颁发者的签名

图2.28 X.509证书结构

(1) 版本号:定义了证书的版本号,这将最终影响证书中包含的信息的类型和格式。目前版本4已颁布,但在实际使用过程中版本3还是占主流。

(2) 证书序列号:序列号是赋予证书的唯一整数值,用于将本证书与同一CA颁发的其他证书区别开来。

(3) 签名算法标识符:该域中含有CA签发证书所使用的数字签名算法的算法标识符,如SHA、RSA等。有CA的签名,便可保证证书拥有者身份的真实性,而且CA也不能否认其签名。

(4) 颁发者名称:该域含有签发证书实体的唯一名称,命名必须符合X.500格式,通常

为某个 CA。

(5) 有效期：每个证书均只能在一个有限的时间段内有效。该有效期表示为两个日期的序列，即起始日期和时间、终止日期和时间，有效期可以短至几秒或长至一世纪。有效期长短取决于许多因素，例如用于数字签名的私钥的使用频率等。

(6) 主体名称：证书拥有者的可识别名称，命名规则使用 X.500 标准，因此在 Internet 中应是唯一的。此字段必须是非空的，除非在扩展项中使用了其他的名字形式。

(7) 主体公钥信息：主体的公钥，同时包括指定该密钥所属公钥密码系统的算法标识符及所有相关的密钥参数。

(8) 颁发者唯一标识符(可选)：证书颁发者唯一标识符，属于可选字段。该字段在实际中很少使用，并且不被 RFC 2459 推荐使用。

(9) 主体唯一标识符(可选)：证书拥有者唯一标识符，属于可选字段，用于不同的实体重用这一证书时标志证书的主体，该字段在实际中很少使用，并且不被 RFC 2459 推荐使用。

(10) 扩展项(可选)：在颁布了 X.509 版本 2 后，人们认为还有一些不足之处，于是提出一些扩展项附在版本 3 证书格式的后面。这些扩展项包括密钥和策略信息、主体和颁发者属性以及证书路径限制。

(11) 颁发者的签名：覆盖了证书的所有其他字段，以及这些字段被 CA 私钥加密后的 Hash 值、签名算法标识等。

2.8.4　公钥基础设施 PKI

公钥证书、证书管理机构、证书管理系统、围绕证书服务的各种软硬件设备以及相应的法律基础共同组成公钥基础设施 PKI(Public Key Infrastructure)。PKI 技术采用证书管理公钥，通过第三方的可信任机构把用户的公钥和用户的其他标识信息(如用户名、e-mail、身份证号、护照号等)绑定在一起，在 Internet 上验证用户的身份。本质上，PKI 是一种标准的公钥密码的密钥管理平台，能够为所有网络应用透明地提供加密、数字签名等服务所需要的密码和证书管理。当前，PKI 以其良好的开放性、安全性和稳定性，已成功应用到了众多领域，在信息安全领域起着越来越重要的作用。

一个有效的 PKI 系统必须是安全的和透明的，用户在获得加密和数字签名服务时，不需要详细了解 PKI 的内部运作机制。一个典型、完整和有效的 PKI 系统应该包含证书的创建和发布以及证书的撤销，一个可用的 PKI 产品还必须提供相应的密钥管理服务，包括密钥的备份、恢复和更新等。没有一个好的密钥管理系统，将极大影响一个 PKI 系统的规模、可伸缩性和在协同网络中的运行成本。

美国是最早推动 PKI 建设的国家，早在 1996 年就成立了联邦 PKI 指导委员会。目前美国联邦政府、州政府、大型企业都建立了 PKI，比较有代表性的主要有 VeriSign 和 Entrust。VeriSign 作为 RSA 的控股公司，借助 RSA 成熟的安全技术，提供了 PKI 产品，为用户之间的内部信息交互提供了安全保障。另外，VeriSign 也提供对外的 CA 服务，包括证书的发布和管理等功能，并且同一些大的生产商，如 Microsoft、Netscape 和 JavaSoft 等保持了伙伴关系，以在 Internet 上提供代码签名服务。

1998 年中国的电信行业也建立了国内第一个行业 CA，此后金融、工商、外贸、海关和一

些省市也建立了自己的行业 CA 和地方 CA。PKI 已经成为世界各国发展电子商务、电子政务、电子金融的基础设施。

1. PKI 的组成和功能

一个典型的 PKI 逻辑结构如图 2.29 所示，其中包括 PKI 策略、软硬件系统、证书机构 CA、注册机构 RA、证书发布系统和 PKI 应用。

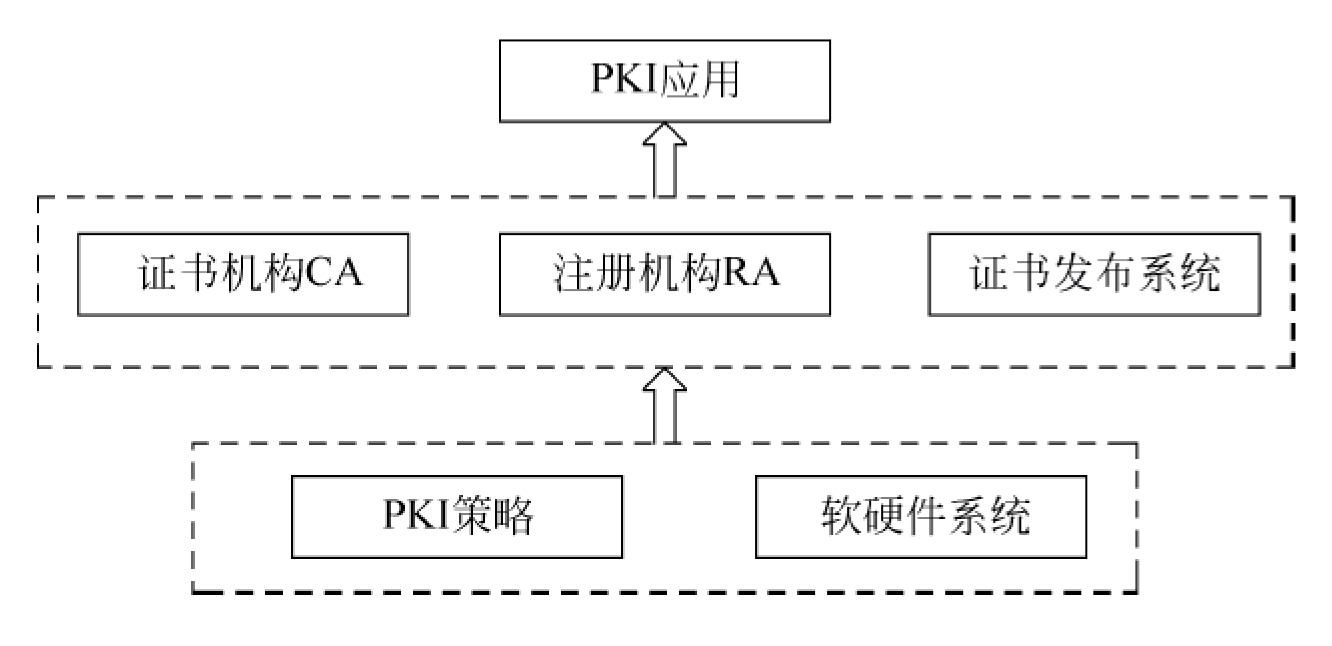


图 2.29 PKI 的逻辑结构

(1) PKI 安全策略：建立和定义了一个组织在信息安全方面的指导方针，同时也定义了密码系统使用的处理方法和原则。

(2) 证书机构 CA：它是整个 PKI 体系中各方都承认的一个值得信赖的公正的第三方机构，是 PKI 的信任基础。它的核心功能是管理公钥的整个生命周期，其作用包括证书的发放、更新、归档等。

(3) 注册机构 RA：主要完成收集用户信息和确认用户身份的功能。这里的用户是指将要向认证中心 CA 申请数字证书的客户，可以是个人，也可以是集团、团体、政府机构等。RA 接受用户的注册申请，审查用户的申请资格，并决定是否同意 CA 给其签发数字证书。需要注意的是，注册机构并不实际给用户签发证书，而只是对用户进行资格审查。

(4) 证书发布系统：负责证书的发放，如可以通过目录服务发放。目录服务器可以是一个组织中现存的，也可以是 PKI 方案中提供的。

(5) 数字证书：在 PKI 中，最重要的信息就是数字证书，可以说，PKI 的所有活动都是围绕数字证书进行的。

(6) PKI 应用：PKI 的应用范围非常广泛，并且在不断发展之中，可以说只要需要使用到公钥的地方就要使用到 PKI，例如安全电子邮件、Web 安全、虚拟专用网等。

一个完整的 PKI 系统的主要功能包括证书颁发、证书废除、证书和 CRL 的公布、证书状态的在线查询等。

(1) 证书颁发：申请者在 CA 的注册机构(RA)进行注册、申请证书。证书的申请可采取在线申请和亲自到 RA 申请两种方式。在线申请就是通过浏览器或其他应用系统通过在线的方式申请证书，这种方式一般用于申请普通用户证书或测试证书。离线方式一般通过人工的方式直接到证书机构的受理点去办理证书申请手续，通过审核后获取证书，这种方式一般用于比较重要的场合，如服务器证书和商家证书等。证书的颁发也可采用两种方式，一种是在线直接从 CA 下载，一种是 CA 将证书制作成介质(IC 卡等)后，由申请者带走。

(2) 证书废除：在 CA 系统中，由于密钥泄露、从属变更、证书终止使用以及 CA 本身私

钥泄密等原因,需要对原来签发的证书进行撤销,证书持有者可以向 CA 申请废除证书。CA 通过认证核实,将证书写入黑名单 CRL(Certificate Revocation List),即证书撤销列表。

(3) 证书和 CRL 的公布。CA 通过轻量级目录访问协议(Lightweight Directory Access Protocol,LDAP)服务器维护用户证书和黑名单(CRL)。它向用户提供目录浏览服务,负责将新签发的证书或废除的证书加入到 LDAP 服务器上。这样用户通过访问 LDAP 服务器就能够得到他人的数字证书或访问黑名单。

(4) 证书状态的在线查询。通常 CRL 签发为一日一次,CRL 的状态同当前证书状态有一定的滞后,证书状态的在线查询向在线证书状态查询协议(Online Certificate Status Protocol,OCSP)服务器发送 OCSP 查询包,包含待验证证书的序列号、验证时戳。OCSP 服务器返回证书的当前状态并对返回结果加以签名。在线证书状态查询比 CRL 更具时效性。

2. PKI 密钥管理

密钥管理是 PKI(主要指 CA)中的一个核心问题,主要包括密钥产生、密钥备份、密钥恢复和密钥更新等。

公钥有两大用途:其一是用于验证数字签名,即消息接收者使用发送者的公钥对消息的数字签名进行验证;其二是用于加密信息,即消息发送者使用接收者的公钥加密用于加密消息的密钥,进行数据加密密钥的传递。相应地,系统中需要配置用于数字签名/验证的密钥对和用于数据加密/解密的密钥对,这里分别称为签名密钥对和加密密钥对。这两对密钥对于密钥管理有不同的要求。

1) 密钥产生

密钥对的产生是证书申请过程中重要的一步,其中产生的私钥由用户保留,公钥和其他信息则交给 CA 中心签名,从而产生证书。依据密钥的使用和系统的安全策略考虑,目前国内外主要有三种实现方法。

(1) 客户自己生成密钥对,然后将公钥以安全的方式传送给 CA,该过程必须保证用户公钥的可验证性和完整性。

(2) CA 替客户生成密钥对,然后将其以安全的方式传送给客户,该过程必须确保密钥对的机密性、完整性和可验证性。该方式下由于客户的私钥为 CA 所知,故对 CA 的可信性有更高的要求。

(3) 由可信的第三方,如密钥管理中心 KMC,生成密钥对,再将相应的密钥传送给 CA 和客户。

在实际运行的 PKI 系统中,根据网络的拓扑结构和安全策略,可采用上述一种或几种方案的组合来实现。

对普通证书和测试证书,一般由客户自己产生密钥,这样产生的密钥强度较小,不适合应用于比较重要的安全网络交易。而对于比较重要的证书,如商家证书和服务器证书等,密钥对一般由专用应用程序或 CA 中心直接产生,这样产生的密钥强度大,适合于重要的应用场合。

另外,根据密钥的应用不同,也可能会有不同的产生方式,例如签名密钥可能在客户端或 RA 中心产生,而加密密钥则需要在 CA 中心直接产生。

2) 密钥的备份和恢复

在一个 PKI 系统中,维护密钥对的备份至关重要,如果没有这种措施,当密钥丢失后,

将意味着加密数据的完全丢失，对于一些重要数据，这将是灾难性的。所以，密钥的备份和恢复也是 PKI 密钥管理中的重要一环。

(1) 签名密钥对

签名密钥对由签名私钥和验证公钥组成。签名私钥具有日常生活中公章、私章的效力，为保证其唯一性，签名私钥绝对不能够作备份和存档，丢失后只需重新生成新的密钥对，原来的签名可以使用旧公钥的备份来验证。验证公钥需要存档，用于验证旧的数字签名。用作数字签名的这一对密钥一般可以有较长的生命周期。

(2) 加密密钥对

加密密钥对由加密公钥和解密私钥组成。为防止密钥丢失时丢失数据，解密私钥应该进行备份，同时还可能需要进行存档，以便能在任何时候解密历史密文数据。加密公钥无须备份和存档，加密公钥丢失时，只需重新产生密钥对。加密密钥对通常用于分发会话密钥，这种密钥应该频繁更换，故加密密钥对的生命周期较短。企业级的 PKI 产品至少应该支持用于加密的安全密钥的存储、备份和恢复。不难看出，这两对密钥的密钥管理要求存在互相冲突的地方，因此，系统必须针对不同的用途使用不同的密钥对，尽管有的公钥体制算法，如目前使用广泛的 RSA，既可以用于加密、又可以用于签名，但在具体使用中仍然必须为用户配置两对密钥、两张证书，其一用于数字签名，其二用于加密。

3) 密钥更新

每一个由 CA 颁发的证书都有有效期，密钥对生命周期的长短由签发证书的 CA 中心来确定，各 CA 系统的证书有效期有所不同，一般为 2～3 年。

当用户的私钥被泄露或证书的有效期快到时，用户应该更新私钥。这时用户可以废除证书，产生新的密钥对，或者申请新的证书。

3. 证书的使用

在实际应用中，为了验证信息的数字签名，用户首先必须获取信息发送者的公钥证书，以及一些额外需要的证书(如 CA 证书等，用于验证发送者证书的有效性)。

证书的获取可以有多种方式，如发送者发送签名信息时附加发送自己的证书，或以另外的单独信息发送证书，或者可以通过访问证书发布的目录服务器来获得，或者直接从证书相关的实体处获得。在一个 PKI 体系中，可以采取某种或某几种上述方式获得证书。

在电子商务系统中，证书的持有者可以是个人用户、企事业单位、商家、银行等。无论是电子商务中的哪一方，在使用证书验证数据时，都遵循同样的验证流程。一个完整的验证过程有以下几步。

(1) 验证者将客户端发来的数据解密。

(2) 将解密后的数据分解成原始数据、签名数据和客户证书三部分。

(3) 用 CA 根证书(CA 的公钥)验证客户证书的签名完整性。

(4) 检查客户证书是否有效(当前时间在证书结构中所定义的有效期内)。

(5) 检查证书是否作废。

(6) 验证客户证书结构中的证书用途。

(7) 用客户的证书(客户的公钥)验证原始数据的签名完整性。

如果以上各项均验证通过，则接收该数据。

4. PKI 的信任模型

最基本的 PKI 结构是单 CA 结构，只需建立一个根 CA，所有用户对此单 CA 信任。这种结构的优点是容易实现，所有用户都能直接相互认证，缺点是不易扩展到支持大量或者不同群体的用户。

在现实中，对于大范围应用，一个 CA 很难得到所有用户的信任并接受它所颁发的数字证书，而且一个 CA 也很难对所有潜在用户有足够全面的了解，因此往往需要多个 CA，这些 CA 之间应该具有某种结构关系，以使不同 CA 之间的证书认证简单方便。

证书用户、证书主体、各个 CA 之间的证书认证关系称为 PKI 的信任模型。人们目前已经提出了多种信任模型。

1）严格层次结构模型

严格层次模型是一个以主从 CA 关系建立的分级 PKI 结构，像一棵倒置的树，如图 2.30 所示。在这个结构中，根 CA 把自己的权力授给多个子 CA，这些子 CA 再将它们的权力授给它们的子 CA，这个过程持续至某个 CA 实际颁发了证书。

该模型的所有实体（包括子 CA 和终端用户）都信任根 CA，因此都必须拥有根 CA 的公钥，这个层次结构按照如下规则建立。

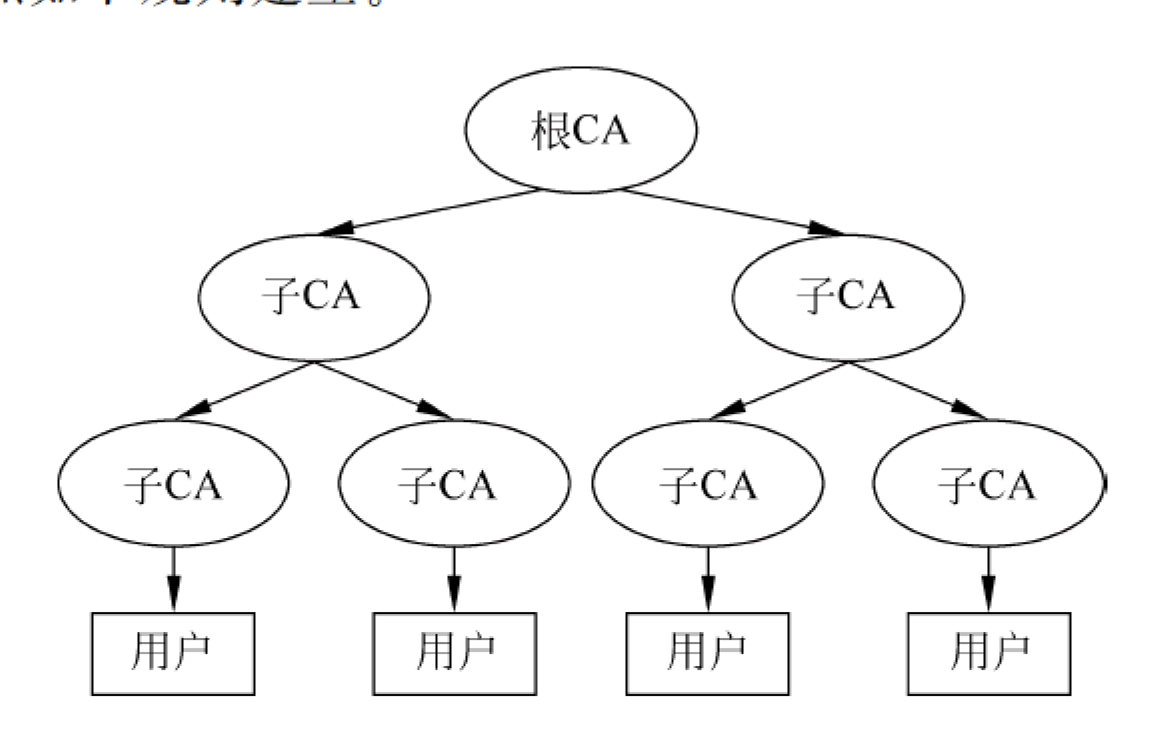


图 2.30　CA 的严格层次模型

(1) 根 CA 认证（为其创建和签署证书）直接在它下面的 CA。

(2) 这些 CA 中的每一个都认证零个或者多个直接在它下面的 CA。

(3) 倒数第二层的 CA 认证终端实体。

任何两个用户之间进行通信，为验证对方的公钥证书，都必须通过根 CA 才能实现。例如一个持有一份可信的根 CA 公钥的终端实体 A 可以通过如下方法检验另一个终端实体 B 的证书。

假设 B 的证书由子 CA3 签发（公钥为 k3），子 CA3 的证书由子 CA2（公钥为 k2）签发，子 CA2 的证书由子 CA1（公钥为 k1）签发，子 CA1 的证书由根 CA（公钥为 k）签发。拥有 k 的终端实体 A 可以利用 k 来验证 CA1 的公钥为 k1，然后利用 k1 来验证 CA2 的公钥为 k2，再利用 k2 来验证 CA3 的公钥为 k3，最终利用 k3 来验证 B 的证书。

建立一个管理全世界所有用户的全球性 PKI 是不现实的。比较可行的办法是各个国家建立自己的 PKI，一个国家之内再建立不同行业或者不同地区的 PKI。但是为了实现跨地区、跨行业甚至跨国际的电子安全业务，这些不同的 PKI 之间互联互通和互相信任是不可避免的。

2）CA 分布式信任结构

分布式信任结构把信任分散到两个或更多个 CA 上。采用严格层次结构的 PKI 系统往往在一个企业或者部门实施，为了将这些 PKI 系统互连起来，可以采用下列两种方式建立。

（1）中心辐射配置：在这种配置中，有一个中心地位的 CA，每个根 CA 都和这个中心 CA 进行交叉认证。

（2）网状配置：所有根 CA 之间进行交叉认证。

在分布式信任结构的中心辐射配置中，中心 CA 并不能被看作是根 CA，如图 2.31 所示。在这个结构中，可能有多个根 CA，每个实体都信任自己的根 CA，他们只拥有自己根 CA 的公钥。

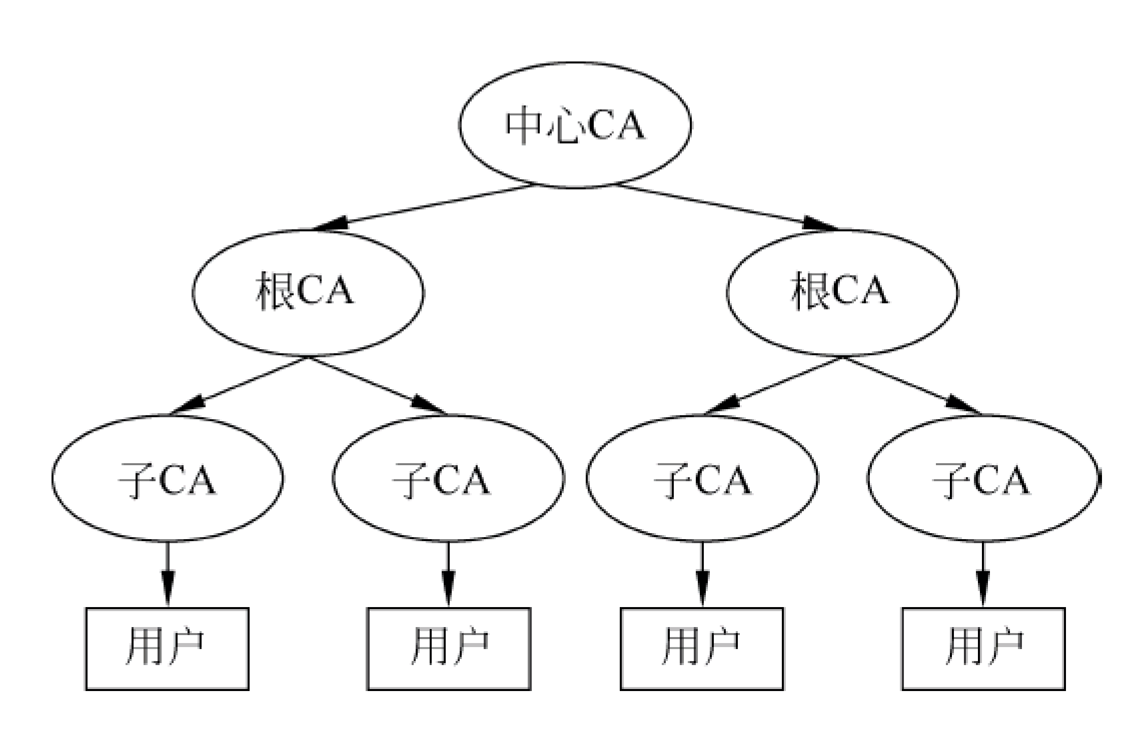


图 2.31　CA 分布式信任结构的中心辐射配置

一个终端实体 A 可以按如下方法检验另一个终端实体 B 的证书。如果他们拥有同一个根 CA 的公钥，认证过程和前面的严格层次结构一样。否则，A 可以利用自己的根 CA 的公钥来验证中心 CA 的公钥，然后利用中心 CA 的公钥来验证 B 的根 CA 的公钥，再利用 B 的根 CA 的公钥向下验证，直至验证终端实体 B 的证书。

3）Web 模型

Web 模型建构在浏览器的基础上，如 Internet Explore 或者 Firefox。浏览器厂商在浏览器中内置了多个根 CA，浏览器的用户最初信任这些 CA 并把它们作为根 CA。这些根 CA 是通过物理嵌入软件来发布的，这样就将 CA 的名字和它的公钥安全绑定。

图 2.32 是预安装在 IE 浏览器中各 CA 的公钥证书，可以通过 IE 浏览器中的“工具→Internet 选项→内容→证书”查看。选中某一个证书，单击“高级”选项，可以看到该证书的各项属性。

4）以用户为中心的信任模型

在以用户为中心的认证模型中，每个用户都直接决定信赖或拒绝哪个证书。最初可信的密钥集可能只有朋友、家人和同事等。这种模型用户可自己决定是否信赖某个证书，安全性和可控性很强。但是由于普通用户很少有了解 PKI 机制的，所以这种模型的适用范围狭窄。这种模型在金融、政府环境都是不适宜的，因为在这些群体中，往往需要以组织的方式控制一些公钥，而不希望完全由用户自己控制。以用户为中心的信任模型最典型的就是著名的安全软件 PGP。

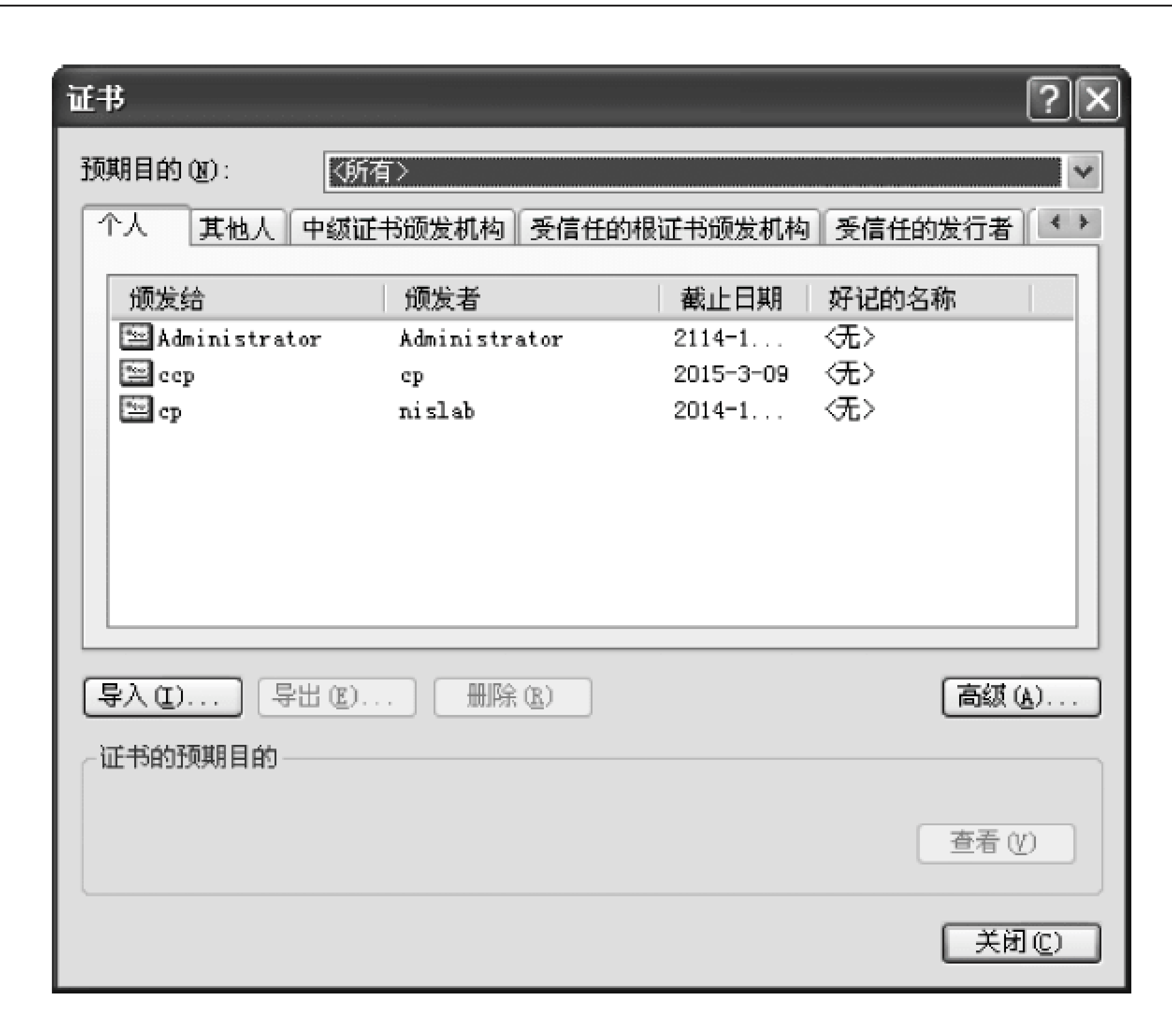


图 2.32　安装在 IE 浏览器中的 CA 公钥证书

5）交叉认证模型

交叉认证模型是一种把各个 CA 连接在一起的机制，可以是单向的，如在 CA 的严格层次结构中，上层 CA 对下层的认证，也可以是双向的，如在分布式信任结构的中心辐射配置中，根 CA 与中心 CA 的相互认证。

在两个 CA 之间的交叉认证是指一个 CA 承认另一个 CA 在一个名字空间中被授权颁发的证书，如图 2.33 所示。

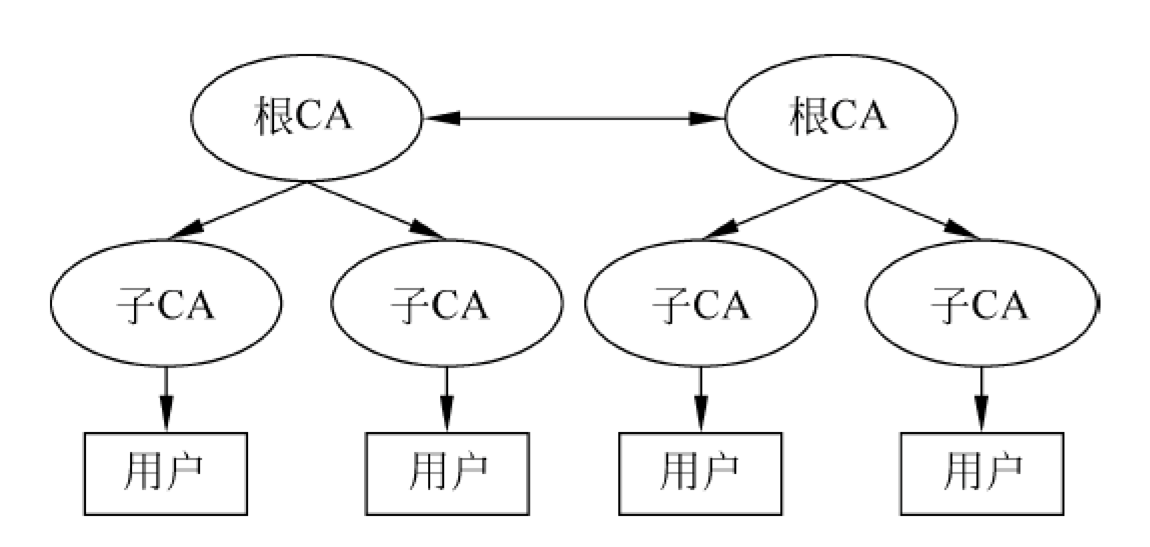


图 2.33　交叉认证模型

例如假设实体 A 已经被 CA1 认证并且拥有 CA1 的公钥 k1，而实体 B 已被 CA2 认证并且拥有 CA2 的公钥 k2。在交叉认证前，A 只能验证 CA1 颁发的证书，而不能验证 CA2 颁发的证书，而 B 则只能验证 CA2 颁发的证书，而不能验证 CA1 颁发的证书。在 CA1 和 CA2 互相交叉认证后，A 就能验证 CA2 的公钥，从而验证 CA2 颁发的证书，B 也能验证 CA1 的公钥从而验证 CA1 颁发的证书。

5. PKI 应用

PKI 技术的广泛应用能满足人们对网络交易安全保障的需求。当然作为一种基础设施，PKI 的应用范围非常广泛，并且在不断发展之中，下面给出几个应用实例。

1）虚拟专用网络（VPN）

VPN 是一种架构在公用通信基础设施上的专用数据通信网络，利用网络层安全协议（尤其是 IPSEC）和建立在 PKI 上的加密和签名技术来获得机密性保护。基于 PKI 技术的

IPSEC 协议现在已经成为架构 VPN 的基础,它可以为路由器之间、防火墙之间或者路由器和防火墙之间提供经过加密和认证的通信。虽然它的实现会复杂一些,但其安全性比其他协议都完善得多。

2）安全电子邮件

随着 Internet 的持续增长,电子邮件的方便快捷已使其成为重要的沟通和交流工具,但目前的电子邮件系统存在较大的安全隐患,主要包括消息和附件可以在不为通信双方所知的情况下被读取、篡改或截掉,发信人的身份无法确认。电子邮件的安全需求也是机密性、完整性、认证和不可否认性,而这些都可以利用 PKI 技术来获得。

目前已经被广泛应用的安全电子邮件协议是 S/MIME(The Secure Multipurpose Internet Mail Extension),这是一个允许发送加密和有签名邮件的协议,该协议的实现需要依赖于 PKI 技术。

3）Web 安全

为了透明地解决 Web 的安全问题,在两个实体进行通信之前,先要建立 SSL 连接,以此实现对应用层透明的安全通信。利用 PKI 技术,SSL 协议允许在浏览器和服务器之间进行加密通信。此外,服务器端和浏览器端通信时双方可以通过数字证书确认对方的身份。结合 SSL 协议和数字证书,PKI 技术可以保证 Web 交易多方面的安全需求,使 Web 上的交易和面对面的交易一样安全。

从目前的发展来说,PKI 的范围非常广,不仅仅局限于通常认为的 CA 机构,它还包括完整的安全策略和安全应用。因此,PKI 的开发也从传统的身份认证到各种与应用相关的应用场合,如企业安全电子商务和政府的安全电子政务等。

另外,PKI 的开发也从大型的认证机构到与企业或政府应用相关的中小型 PKI 系统发展,既保持了兼容性,又和特定的应用相关。

2.9 小　　结

密码学是关于加密和解密变换的一门科学,是保护数据和信息的有力武器,密码学作为信息安全理论与技术的基石,在信息安全领域中发挥着中流砥柱的作用。通过加密可以实现信息的机密性,基于密码技术的消息认证还能提供信息的完整性检验,即提供一种当某些信息被修改时可被用户检验出的机制;密码技术还可实现数字签名,防止通信双方的相互抵赖。现代密码体制主要分为对称密码体制和公钥密码体制,密码体制的安全性取决于密钥的安全,密钥管理至关重要。对于公钥密码体制需要基于公钥基础设施 PKI 实现公钥的有效安全管理。

习　　题

一、填空题

1. 现代密码理论的理论基础是香农于 1949 年发表的著名论文________,对称密码体制和非对称密码体制的代表算法分别是________和________。

2. 单密钥系统的加密密钥和解密密钥________或________。

3. 公钥密码体制的编码系统是基于数学中的________函数。使用双密钥系统的公开密钥对消息进行变换可以实现________,使用私有密钥对消息进行变换可以实现________。

4. 香农提出设计密码系统的两种基本方法是________和________。

5. 消息认证是实现________的主要手段。

6. 根据明文处理方式和密钥使用方式的不同,可以将密码体制分成分组密码和________。

7. Kerckhoffs 准则指出数据的安全应该基于________的保密。

8. 破译密码的方法有两类:________和________。

9. 一个密码仅当它能经受得住________攻击才是可取的。

10. ________和________是密码体制常用的基本技术。

二、选择题

1. 数据加密标准 DES 采用的密码类型是(　　)。
   A. 序列密码　B. 分组密码　C. 散列码　D. 随机码

2. 数字签名技术不能解决的安全问题是(　　)。
   A. 第三方冒充　B. 接收方篡改　C. 信息窃取　D. 接收方伪造

3. 关于 CA 和数字证书的关系,以下说法不正确的是(　　)。
   A. 数字证书是保证双方之间的通信安全的电子信任关系,它由 CA 签发
   B. 数字证书一般依靠 CA 中心的对称密钥机制来签名
   C. 在电子交易中,数字证书可以用于表明参与方的身份
   D. 数字证书能以一种不能被假冒的方式证明证书持有人身份

4. 若 A 给 B 发送一封邮件,并想让 B 能验证邮件是由 A 发出的,则 A 应该选用(　　)对邮件加密。
   A. A 的公钥　B. A 的私钥　C. B 的公钥　D. B 的私钥

5. 在 RSA 密码体制中,如果 $p=3$,$q=7$,取 $e=5$,根据这些已知条件,可得密钥为(　　)。
   A. {5,12}　B. {5,21}　C. {17,21}　D. 以上都不是

6. PKI 的主要组成不包括(　　)。
   A. CA　B. IPSEC　C. RA　D. CR

7. 数字签名要预先使用单向 Hash 函数进行处理的原因是(　　)。
   A. 多一道加密工序使密文更难破译
   B. 提高密文的计算速度
   C. 缩小签名密文的长度,加快数字签名和验证签名的运算速度
   D. 保证密文能正确还原成明文

8. 对散列函数最好的攻击方式是(　　)。
   A. 穷举攻击　B. 中间人攻击　C. 字典攻击　D. 生日攻击

三、简答题

1. 什么是密码学?什么是密码编码学和密码分析学?

2. 现代密码系统的 5 个组成部分是什么?

3. 密码分析主要有哪些形式?各有何特点?

4．对称密码体制和非对称密码体制各有何优缺点？

5．假设 Alice 想发送消息 M 给 Bob，出于机密性以及确保完整性和不可否认性的考虑，Alice 可以在发送前对消息进行签名和加密，其操作顺序对安全性有无影响？

6．简述用 RSA 算法实现机密性、完整性和抗否认性的原理。

7．已知有明文“public key encryption”，先将明文以 2 个字母为组分成 10 块，如果利用英文字母表的顺序，即 a＝00，b＝01，…，将明文数据化。现在令 $p=53$，$q=58$，请计算出 RSA 的加密明文。

8．在使用 RSA 公钥中如果截取了发送给其他用户的密文 $C=10$，若此用户的公钥为 $e=5$、$n=35$，请问明文的内容是什么？

9．什么是散列函数？散列函数有哪些应用？

10．消息认证的方法有哪些？

11．什么是数字签名？常用的算法有哪些？

12．请谈谈你对密码体制无条件安全和计算上安全的理解。

13．知识扩展：访问网站 http://www.tripwire.com，了解完整性校验工具 Tripwire 的更多信息。

14．知识扩展：访问中国电子签名网站 http://www.eschina.info，了解电子签名的研究动态和最新应用。

15．知识扩展：访问国家密码管理局（国家商用密码管理办公室）网站 http://www.oscca.gov.cn，了解我国商用密码管理规定、商用密码产品等信息。

第3章 信息系统的物理安全和可靠性

计算机硬件及其运行环境是计算机信息系统运行的基础,它们的安全直接影响着整个信息系统的安全。由于自然灾害、设备自身的缺陷、设备的自然损坏和受到环境干扰等自然因素,以及人为的窃取和破坏等原因,计算机设备和其中信息的安全面临很大的问题。本章主要讨论从物理层面增强信息系统安全的方法。

3.1节给出了物理安全的定义,指出了狭义和广义物理安全包含范畴的不同,明确本章讲述的物理安全包括环境安全、设备安全、媒体(介质)安全、系统安全。3.2~3.4节分别介绍了环境安全、设备安全和媒体(介质)安全。3.5节介绍了系统安全,可靠性是评价系统安全的重要指标,指出提高系统可靠性一般采取避错、容错和容灾备份技术。3.6、3.7节分别介绍了容错和灾难恢复技术。

3.1 物理安全概述

根据国家标准《信息安全技术信息系统物理安全技术要求》,物理安全(Physical Security)是指为了保证信息系统安全可靠运行,确保信息系统在对信息进行采集、处理、传输、存储的过程中,不致受到人为或自然因素的危害而使信息丢失、泄露或破坏,对计算机设备、设施(包括机房建筑、供电、空调等)、环境人员、系统等采取适当的安全措施。

物理安全是计算机网络信息系统运行的基础,直接影响着计算机信息系统的安全。以下是计算机系统物理安全遭到破坏的一个典型的例子。

2006年12月26日晚8时26分至40分间,我国台湾屏东外海发生地震。大陆出口光缆、中美海缆、亚太1号等至少6条海底通信光缆发生中断,造成我国大陆至台湾地区、美国、欧洲的通信线路大量中断,互联网大面积瘫痪,除我国外,日本、韩国、新加坡网民均受到影响。

传统意义的物理安全包括设备安全、环境安全以及介质安全,涉及到的安全技术解决了由于设备/设施/介质的硬件条件所引发的信息系统物理安全威胁问题,从系统的角度看,这一层面的物理安全是狭义的物理安全,是物理安全的最基本内容。广义的物理安全还应包括由软件、硬件、操作人员组成的整体信息系统的物理安全,即包括系统物理安全。

本章讨论的物理安全包括环境安全、设备安全、介质安全、系统安全4个方面。

1. 环境安全

环境安全是指为保证信息系统安全可靠运行所提供的安全运行环境,使信息系统得到物理上的严密保护,从而降低或避免各种安全风险。技术要素包括机房场地选择、机房屏蔽、防火、防水、防雷、防鼠、防盗防毁、供配电系统、空调系统、综合布线、区域防护等方面。

2. 设备安全

设备安全是指为保证信息系统的安全可靠运行,降低或阻止人为或自然因素对硬件设

备安全可靠运行带来的安全风险，对硬件设备及部件所采取的适当安全措施，其技术要素包括设备的防盗、防电磁泄露、抗电磁干扰、电源保护以及设备振动、碰撞、冲击适应性等方面。

3. 介质安全

介质安全是指存储信息的介质的安全，能够安全保管、防盗、防损坏和防霉。

4. 系统安全

系统安全是指为保证信息系统的安全可靠运行，降低或阻止人为或自然因素从物理层面对信息系统保密性、完整性、可用性带来的安全威胁，从系统的角度采取的适当安全措施，如通过边界保护、配置管理、设备管理等措施保护信息系统的保密性；通过容错、故障恢复、系统灾难备份等措施确保信息系统的可用性；通过设备访问控制、边界保护、设备及网络资源管理等措施确保信息系统的完整性。

3.2 环境安全

3.2.1 环境安全面临的威胁

计算机的运行环境对计算机的影响非常大，影响计算机运行的环境因素主要有温度、湿度、灰尘、腐蚀等，这些因素从不同侧面影响计算机的可靠工作。

1. 温度

无论是台式机还是笔记本，计算机元器件如CPU、主板、显卡、声卡、网卡都是封闭在机箱内的，计算机在工作的时候，机箱内部温度很高，所以计算机都配备有风扇等散热设备，但是如果计算机持续工作或外部环境过高，计算机元器件的温度会过高，即使有散热设备也无法保证计算机处于正常工作的温度范围。计算机正常工作的温度范围是0～45℃。当环境温度超过60℃时，计算机系统就不能正常工作，温度每升高10℃，电子元器件的可靠性就会降低25%。元器件可靠性降低会直接影响计算机的正确运算，从而影响计算结果的正确性。

另外，温度对磁介质的导磁率影响很大，磁盘表面的磁介质具有热胀冷缩的特性，如果温度过高或过低，磁盘表面会发生变形，从而造成数据的读写错误；温度过高还会使插头、插座、计算机主板、各种信号线腐蚀速度加快，容易造成接触不良；温度过高也对显示器造成不良的影响，会使显示器各线圈骨架尺寸发生变化，使图像质量下降。

总之，环境温度过高或过低都容易引起硬件损坏，计算机工作的环境温度一般应控制在20℃左右。

2. 湿度

如果环境相对湿度低于40%，环境比较干燥；如果高于60%，则比较潮湿。湿度过高或过低对计算机的可靠运行都有影响。

湿度过大会使元器件的表面附着一层很薄的水膜，造成元器件各引脚之间的漏电。当水膜中含有杂质时，它们会附着在元器件引脚、导线、接头表面，造成这些表面发霉和触点腐蚀。磁性介质是多孔材料，在相对湿度高的情况下，它就会吸收空气中的水分变潮，使其导磁率发生明显变化，造成磁介质上的信息读写错误。

湿度过低则意味着环境比较干燥，过于干燥就很容易产生静电。在环境非常干燥的情况下去触摸元器件，会造成元器件的损害。除此之外，过于干燥的空气还可能会造成磁介质上的信息被破坏、纸张变脆、印刷电路变形等危害。

计算机正常的工作湿度应该控制在40%～60%。

3. 灰尘

空气中的灰尘对计算机中的精密机械装置，如磁盘、光盘驱动器影响很大。磁盘机、光盘机的读头与盘片之间的距离很小，不到1μm，在高速旋转过程中各种灰尘，其中包括纤维性灰尘会附着在盘片表面，当读头靠近盘片表面读信号的时候，就可能擦伤盘片表面或者磨损读头，造成数据读写错误或数据丢失。灰尘中还可能有导电性和腐蚀性尘埃，附着在元器件与电子线路的表面，在湿度很大的情况下，会造成短路或腐蚀裸露的金属表面。因此需要对进入机房的空气进行过滤，并采取严格的机房卫生制度，降低机房灰尘的含量。

3.2.2 环境安全防护

为规范电子信息系统机房设计，确保电子信息系统设备安全、稳定、可靠地运行，GB 50174-2008《电子信息系统机房设计规范》(以下简称《规范》)对机房分级与性能要求、机房位置与设备布置、环境要求、建筑与结构、空气调节、电气、电磁屏蔽、机房布线、机房监控与安全防范、给水排水、消防等方面提出了具体要求。

1. 机房安全等级

计算机系统中的各种数据依据其重要性和保密性，可以划分为不同等级，需要提供不同级别的保护。对于高等级数据采取低水平的保护会造成不应有的损失，对不重要的信息提供多余的保护，又会造成不应有的浪费。因此，应对计算机机房规定不同的安全等级。《规范》将电子信息系统机房划分为A、B、C三级，设计时应根据机房的使用性质、管理要求及其在经济和社会中的重要性确定所属级别。

符合下列情况之一的电子信息系统机房应为A级：

(1) 电子信息系统运行中断将造成重大的经济损失。

(2) 电子信息系统运行中断将造成公共场所秩序严重混乱。

例如国家气象台、国家级信息中心、重要的军事指挥部门、大中城市的机场、广播电台、电视台等的电子信息系统机房和重要的控制室应为A级。

符合下列情况之一的电子信息系统机房应为B级：

(1) 电子信息系统运行中断将造成较大的经济损失。

(2) 电子信息系统运行中断将造成公共场所秩序混乱。

例如，科研院所、高等院校、三级医院、大中城市的气象台、省部级以上政府办公楼、大型工矿企业等的电子信息系统机房和重要的控制室应为B级。

不属于A级或B级的电子信息系统机房为C级。

A级电子信息系统机房内的场地设施应按容错系统配置，在电子信息系统运行期间，场地设施不应因操作失误、设备故障、外电源中断、维护和检修而导致电子信息系统运行中断。容错系统是具有两套或两套以上相同配置的系统，在同一时刻，至少有两套系统在工作。按容错系统配置的场地设备，至少能经受住一次严重的突发设备故障或人为操作失误

事件而不影响系统的运行。

B级电子信息系统机房内的场地设施应按冗余要求配置，在系统运行期间，场地设施在冗余能力范围内，不应因设备故障而导致电子信息系统运行中断。冗余系统是重复配置系统的一些部件或全部部件，当系统发生故障时，冗余配置的部件介入并承担故障部件的工作，由此减少系统的故障时间。

C级电子信息系统机房内的场地设施应按基本需求配置，在场地设施正常运行的情况下，应保证电子信息系统运行不中断。

2. 机房位置及设备布置要求

1）机房位置选择

电子信息系统机房位置选择应符合下列要求：

（1）电力供给应稳定可靠，交通、通信应便捷，自然环境应清洁。

（2）应远离产生粉尘、油烟、有害气体以及生产或储存具有腐蚀性、易燃、易爆物品的场所。

（3）远离水灾、火灾隐患区域。

（4）远离强振源和强噪声源。

（5）避开强电磁场干扰。

对于多层或高层建筑物内的电子信息系统机房，在确定主机房的位置时，应对设备运输、管线敷设、雷电感应和结构荷载等问题进行综合考虑和经济比较；采用机房专用空调的主机房，应具备安装室外机的建筑条件。

2）机房组成

电子信息系统机房的组成应根据系统运行特点及设备具体要求确定，一般宜由主机房、辅助区、支持区和行政管理区等功能区组成。

主机房的使用面积应根据电子信息设备的数量、外形尺寸和布置方式确定，并预留今后业务发展需要的使用面积。辅助区的面积宜为主机房面积的0.2～1倍；用户工作室可按每人3.5～4m^{2}计算；硬件及软件人员办公室等有人长期工作的房间，可按每人5～7m^{2}计算。

3）设备布置

电子信息系统机房的设备布置应满足机房管理、人员操作和安全、设备和物料运输、设备散热、安装和维护的要求。

产生尘埃及废物的设备应远离对尘埃敏感的设备，并宜布置在有隔断的单独区域内。

当机柜或机架上的设备为前进风/后出风方式冷却时，机柜和机架的布置宜采用面对面和背对背的方式。

主机房内和设备间的距离应符合下列规定：

（1）用于搬运设备的通道净宽不应小于1.5m；

（2）面对面布置的机柜或机架正面之间的距离不应小于1.2m；

（3）背对背布置的机柜或机架背面之间的距离不应小于1m；

（4）当需要在机柜侧面维修测试时，机柜与机柜、机柜与墙之间的距离不应小于1.2m；

（5）成行排列的机柜，其长度超过6m时，两端应设有出口通道；当两个出口通道之间的距离超过15m时，在两个出口通道之间还应增加出口通道；出口通道的宽度不应小于1m，局部可为0.8m。

3. 机房的环境条件

1）温度、湿度及空气含尘浓度

主机房和辅助区内的温度、相对湿度应满足电子信息设备的使用要求；无特殊要求时，应根据电子信息系统机房的等级，按照表 3.1 所示的要求执行。

表 3.1　机房温度、相对湿度要求

项　　目	技术要求			备注
	A 级	B 级	C 级	
主机房温度（开机时）	23℃±1℃		18～28℃	不得结露
主机房相对湿度（开机时）	40%～55%		35%～75%	
主机房温度（停机时）	5～35℃			
主机房相对湿度（停机时）	40%～70%		20%～80%	不得结露
主机房和辅助区温度变化率（开、停机时）	<5℃/h		<10℃/h	
辅助区温度、相对湿度（开机时）	18～28℃、35%～75%			
辅助区温度、相对湿度温度（停机时）	5～35℃、20%～80%			
不间断电源系统电池室温度	15～25℃			

A 级和 B 级主机房的含尘浓度，在静态条件下测试，每升空气中大于或等于 0.5μm 的尘粒数应少于 18 000 粒。

对于重要的系统机房，应安装吹尘、吸尘设备，排除进入人员所带的灰尘。空调系统进风口应安装空气滤清器，并应定期清洁和更换过滤材料，以防灰尘进入，同时进风压力要大，房间要密封，使室内空气压力高于室外，防止室外灰尘进入室内。

2）噪声、电磁干扰、振动及静电

有人值守的主机房和辅助区，在电子信息设备停机时，在主操作员位置测量的噪声值应小于 65dB(A)。

主机房内无线电干扰场强，在频率为 0.15～1000MHz 时，主机房和辅助区内的无线电干扰场强不应大于 126dB。

主机房和辅助区内磁场干扰环境场强不应大于 800A/m。

在电子信息设备停机条件下，主机房地板表面垂直及水平向的振动加速度值，不应大于 $500mm/s^2$。

主机房和辅助区的绝缘体的静电电位不应大于 1KV。

机房场地环境要求更详细的内容，可以参阅 GB 50174-2008《电子信息系统机房设计规范》。

3.3　设 备 安 全

3.3.1　设备安全面临的威胁

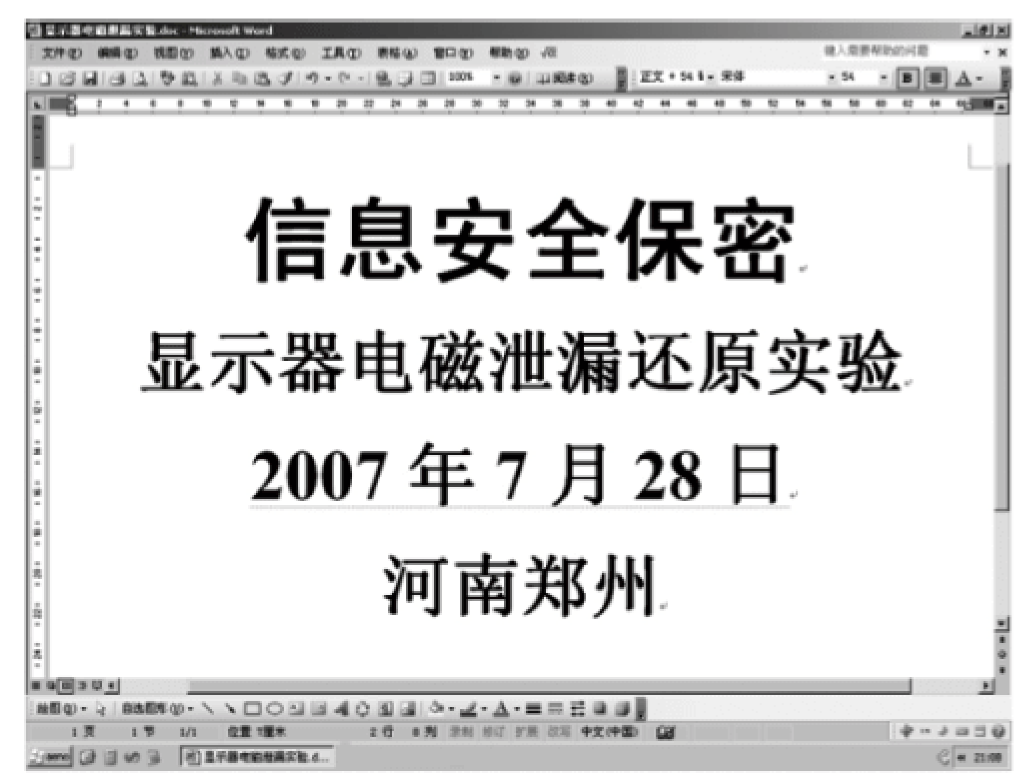
1. 计算机硬件容易被盗

纵观 PC 的发展历史，微型化、移动化是其发展趋势。人们最早使用的是 CRT 显示器和较大的机箱，设备非常笨重，目前 CRT 显示器已经被液晶显示器取代；便携式电脑、以

iPad 为代表的智能移动终端的出现给人们的工作和娱乐带来了方便。PC 朝着体积越来越小、越来越便携的方向发展,给人们带来便利的同时也带来了容易被盗窃的风险。目前,PC 的机箱一般都设计成便于用户打开的,有的甚至连螺钉旋具也不需要,而便携式电脑、智能移动终端整个机器都能够很容易被搬走,其中数据的安全就更谈不上了。

2. 电磁泄露

电磁泄露是指电子设备的杂散(寄生)电磁能量通过导线或空间向外扩散,使用专门的接收设备将这些电磁辐射接收下来,经过处理,就可以恢复还原出原信息,如图 3.1 所示。任何处于工作状态的电磁信息设备如主机、磁盘、显示器、打印机等工作时都会产生不同程度的电磁泄露,尤其是显示器,由于显示的信息是给人阅读的,不加任何保密措施,因此其产生的辐射是最容易造成泄密的。随着信息技术设备处理速度的不断提高,电磁发射的强度也不断增强,对信息设备安全的威胁也就越大。



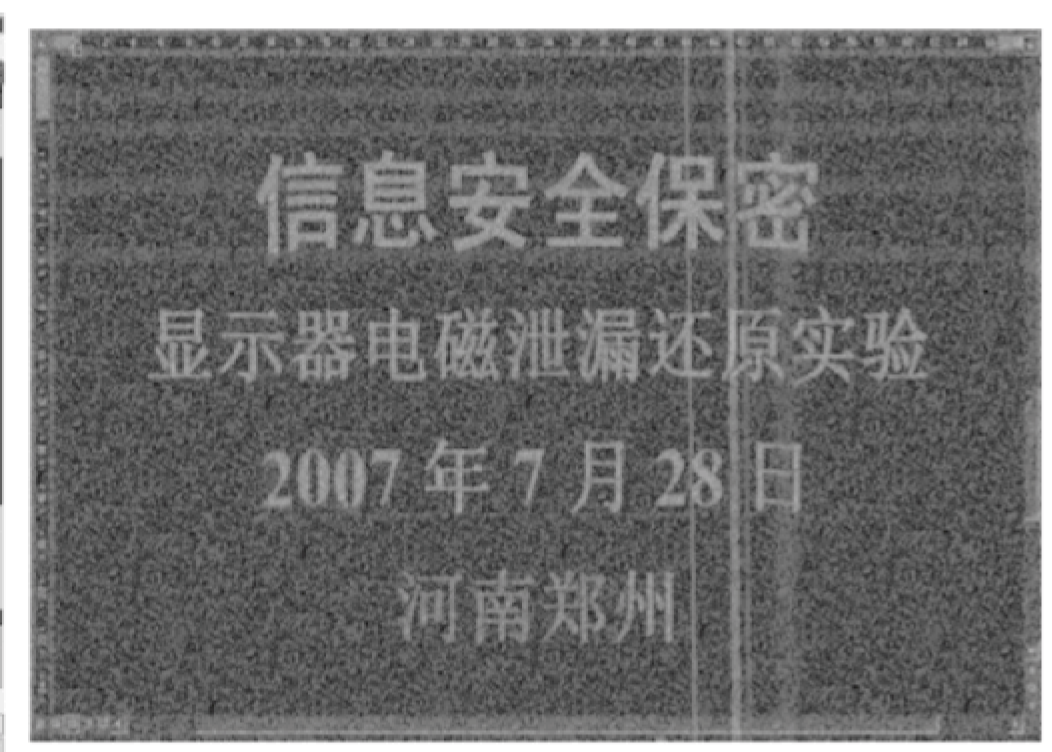


图 3.1 电磁泄露的还原效果

计算机及其外部设备的信息可以通过两种方式泄露出去。一种是以电磁波的形式辐射出去,称为辐射泄露。经实际仪器测试,在距离计算机几百米以外的距离可以根据接收到的电磁波复现显示器上显示的信息,计算机屏幕上的信息在其所有者毫不知晓的情况下泄露出去。1985 年,在法国召开的“计算机与通信”国际会议上,荷兰的一位工程师 WinvanEck 公开了他窃取微机信息的技术。他用价值仅几百美元的器件对普通电视机进行改造,然后装在汽车里,从楼下的街道接收到了放置在 8 层楼上的计算机电磁泄漏的信息,并显示出计算机屏幕上显示的图像。另一种是通过各种线路和金属管传导出去的,称为传导泄露。例如,计算机的电源线、机房内的电话线、上(下)水管道和暖气管道、地线等都可能作为传导介质。这些金属导体有时也起着天线作用,将传导的信号辐射出去。在这些泄漏源中,最大量和最基本的辐射源是载流导线。美国曾在 20 世纪 70 年代在前苏联领海纵深内部的鄂霍次克海 120m 深的海底军事通信电缆上安装了 6m 长的窃听设备,记录了所有经过电缆的通信信号,由于没有采取任何加密措施,大量的军事情报便轻而易举地落在了美国人的手里。

理论分析和实际测量表明,影响计算机电磁辐射强度的因素有:(1)功率和频率。设备的功率越大,辐射强度越大。信号频率越高,辐射强度越大。(2)距离因素。在其他条件相同的情况下,离辐射源越近,辐射强度就越大;离辐射源越远,则辐射强度越小。也就是说,辐射强度与距离成反比。(3)屏蔽状况。辐射源是否屏蔽,屏蔽情况的好坏,对辐射强度的

影响都很大。

3. 电气与电磁干扰

电气干扰是指电网电压引起的干扰，常见的电气干扰是指电压瞬间较大幅度的变化、突发的尖脉冲或电压不足甚至掉电。例如，机房内使用较大功率的吸尘器、电钻，机房外使用电锯、电焊机等大用电量设备，这些情况都容易在附近的计算机电源中产生电器噪声信号干扰。这些干扰一般容易破坏信息的完整性，有时还会损坏计算机设备。防止电气干扰的办法是采用稳压电源或不间断电源，为了防止突发的电源尖脉冲，对电源还要增加滤波和隔离措施。

电磁干扰是指经辐射或传导的电磁能量对设备或信号传输造成的不良影响。过去人们往往认为计算机是具有逻辑特征的数字系统，受电磁干扰的影响不大，但随着微电子技术的发展，计算机已朝高速度、高灵敏度、高集成度的方向发展，使得系统的抗电磁干扰度降低。比较常见的一种现象就是站在电视机前或计算机前使用手机时，计算机中会出现波形，这就是电磁干扰。

一方面计算机本身会产生电磁干扰。计算机中的元器件长期使用后其性能会衰减，它们的性能参数往往会偏离理论值，加之工作环境温度不稳定，引起电子线路、设备或系统内部元器件参数改变，从而使元器件存在不同程度的噪声干扰；每个元器件和每根导线上均流过一定大小的电流，因此其周围都会形成一定大小的磁场。当计算机电路中的元器件或线路布局不合理、电路间耦合不良时，就会在导线间产生分布电容或电感，寄生耦合便通过它们耦合进计算机，使信号畸变出错；如果信号线阻抗与负载阻抗不完全匹配，脉冲信号就会在传输线中产生反射现象，使信号波形产生瞬时冲击，造成电路逻辑故障。计算机插件印制板金属化孔通导不良、印制线粗细不均匀，都会产生信号反射干扰；计算机中的高频电路不仅会产生时序信号，还会产生辐射干扰。计算机内部产生的电磁干扰不但会造成计算机本身的工作异常，而且还可能造成计算机数据信息的失密和失窃。

另一方面计算机外部的设备也会产生电磁干扰。计算机工作在一段很宽的工作频率范围内，它基本上与工业、科技、医学高频设备、广播、电视、通信、雷达等射频设备的工作频段相同，致使计算机工作在一个相当复杂的电磁环境中，容易使这些设备受干扰。如果在计算机房内使用较大功率的吸尘器、电钻，在机房外使用电锯、电焊机等大用电量设备，这些设备的使用会对计算机信号造成干扰，甚至会造成传输信息的丢失、计算机设备的破坏。来自于大自然的雷电、大气放电、地球热辐射的干扰会产生随机电流，轻则增加电噪声干扰，使计算机信息出错，重则使计算机元器件击穿，使计算机设备损坏；静电危害是计算机、半导体器件的“大敌”，是造成微机半导体损坏的基本原因。有研究指出，当穿塑料鞋走动，穿尼龙或丝绸工作服在工作台前长期工作时都可能产生很高的静电电压，不仅会使磁记录破坏，还会使计算机设备外壳产生静电感应。

因此外部电磁环境的干扰和系统内部的互相干扰，严重威胁着计算机系统工作的稳定性和可靠性。

3.3.2 设备安全防护

设备安全防护包括设备的防盗、防止电磁泄漏、抗电磁干扰、电源保护等。

1. 设备防盗

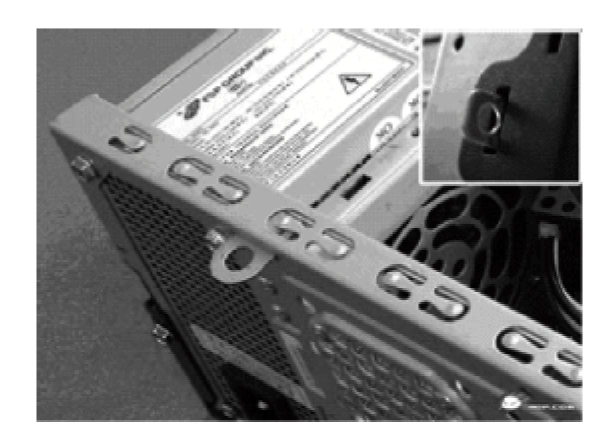
设备防盗就是利用一定的防盗手段保护计算机信息系统的设备和部件，以提高计算机

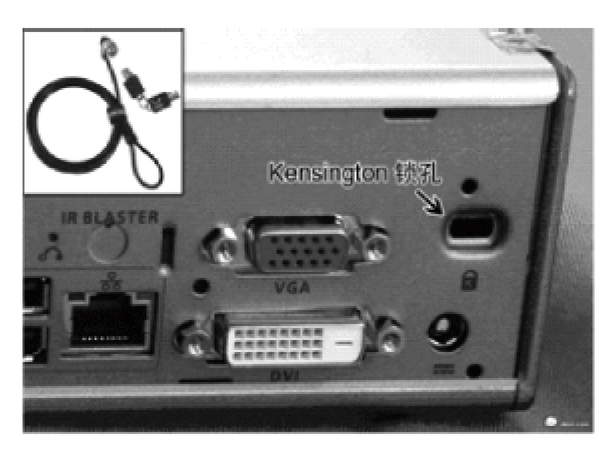
信息系统设备和部件的安全性。早期的防盗主要采取增加质量或胶粘的方法，使设备长久固定或黏接在一个固定点。虽然增加了安全性，但对于移动和调整位置十分不便。之后，又出现了将设备和固定盘用锁连接，打开锁才能搬运设备的方法。常见的锁有机箱锁扣、Kensington 锁孔、机箱电磁锁等。

机箱锁扣的实现方式非常简单。在机箱上固定一个带孔的金属片，然后在机箱侧板上打一个孔，当侧板安装在机箱上时，金属片刚好穿过锁孔，此时用户在锁孔上加装一把锁就实现了防护功能。这种锁实现起来比较简单、造价低，但防护强度有限、安全系数低。

Kensington 锁孔由美国的 Kensington 公司发明，因此而得名。Kensington 锁孔需要配合 Kensington 线缆锁来实现防护功能。使用时将钢缆的一头固定在桌子或其他固定装置上，另一头将锁头固定在机箱上的 Kensington 锁孔内，就实现了防护功能。其特点是固定方式灵活，对于一些开在机箱侧板上的 Kensington 锁孔，不仅可以锁定机箱侧板，而且钢缆还能防止机箱被人挪动或搬走。

机箱电磁锁主要应用于高端商用 PC 产品上，实现方式是将电磁锁安装在机箱内，嵌入在 BIOS 中的子系统通过密码实现电磁锁的开关管理。这种防护方式更加安全和美观，也是一种人性化的安全防护方式，如图 3.2 所示。


(a) 机箱锁扣



(b) Kensington锁孔


(c) 机箱电磁锁

图 3.2 固定盘用锁

另外还有一种使用光纤电缆保护设备的方法，这种方法是将光纤电缆连接到每台重要的设备上，光束沿光纤传输，如果通道受阻则报警。这种保护装置比较简单，一套装置可以保护机房内所有的重要设备，并且设备还可以随意移动、搬运。

一种更方便的方法是使用智能网络传感设备。将传感设备安放在机箱边缘，当机箱盖被打开时，传感开关自动复位，此时传感开关通过控制芯片和相关程序，将此次开箱事件自动记录到 BIOS 中或通过网络及时传给网络设备管理中心，实现集中管理。智能网络传感设备是一种创新的防护方式，但对电源和网络的依赖性大。如果在关掉电源和切断网络的情况下打开机箱，传感器是无法捕获到的。

另外，安装视频监视系统也是必不可少的，视频监视系统是一种更为可靠的防护设备，能对系统运行的外围环境、操作环境实施监控。对重要的机房，还应采取特别的防盗措施，如值班守卫，出入口安装金属防护装置保护安全门、窗户。

2. 防电磁泄漏

计算机是一种非常复杂的机电一体化设备，工作在高速脉冲状态的计算机就像是一台很好的小型无线电发射机和接收机，不但产生电磁辐射泄漏保密信息，而且还可以引入电磁干扰影响系统的正常工作。尤其是在微电子技术和卫星通信技术飞速发展的今天，计算机电磁辐射泄密的危险越来越大。国际上把信息辐射泄漏技术简称为 TEMPEST(Transient

ElectroMagnetic Pulse Emanations Standard Technology,瞬时电磁脉冲发射标准技术),这种技术主要研究与解决计算机和外部设备工作时因电磁辐射和传导产生的信息外漏问题,具体研究内容包括电子设备辐射的途径与方式、对电子信息设备辐射泄漏如何防护、如何从辐射信息中提取有用信息、信息辐射的测试技术与测试标准。

计算机设备的防泄漏措施主要有屏蔽技术、使用干扰器、滤波技术、采用低辐射设备、隔离和合理布局等。

1) 屏蔽技术

屏蔽是 TEMPEST 技术中的一项基础措施。屏蔽最典型的例子就是电梯,电梯提供了一个屏蔽的环境,屏蔽的效果是在电梯中手机接收不到信号。根据不同的需要,屏蔽方法包括整体屏蔽、设备屏蔽和元器件屏蔽。整体屏蔽的方法是采用金属网把需要保护的房间屏蔽起来,为了保证良好的屏蔽效果,金属网接地要良好,并且要经过严格的测试验收。整体屏蔽技术适用于需要处理高度保密信息的场合,如军、政首脑机关的信息中心和驻外使馆等地方,应该将信息中心的机房整个屏蔽起来。整体屏蔽的费用比较高,出于对成本的控制和保密性要求的降低,也可以将设备屏蔽,把需要屏蔽的计算机和外部设备放在体积较小的屏蔽箱内,该屏蔽箱要很好地接地,对于从屏蔽箱内引出的导线也要套上金属屏蔽网。对于电子线路中的局部元器件如 CPU、内存条等强辐射部件可采用屏蔽盒进行屏蔽。

2) 使用干扰器

干扰器是一种能辐射电磁噪声的电子仪器,它通过增加电磁噪声降低辐射泄露信息的总体信噪比,从而增大辐射信息被截获后破解还原的难度,达到“掩盖”真实信息的目的。具体的方法是将一台能产生噪声的干扰器放置在计算机设备的旁边,干扰器产生的噪声与计算机设备产生的信息辐射一起向外泄露。

干扰技术可分为白噪声干扰技术和相关干扰技术两种。白噪声干扰技术的原理是使用白噪声干扰器发出强于计算机电磁辐射信号的白噪声,起到阻碍和干扰接收的作用。这种方法有一定的作用,但由于要靠掩盖的方式进行干扰,发射的功率又必须足够强,所以会造成控件的电磁污染,而白噪声干扰也容易被接收方使用较为简单的方法进行滤除或抑制解调接收。相关干扰技术的原理是使用相关干扰器发出能自动跟踪计算机电磁辐射信号的相关干扰信号,使电磁辐射信号被扰乱,起到乱数加密的效果,使接收方即使接收到电磁辐射信号也无法调节出信号所携带的真实信息,如图 3.3 所示。相对于白噪声干扰技术,相关干扰技术对环境的电磁污染较小,且使用简单、效果显著,比较适合于单独工作的个人计算机上。



图 3.3　相关干扰器

3) 滤波技术

滤波能非常有效地减少和抑制电磁泄露,是抑制传导泄露的主要方法之一,主要方法是在信号传输线、公共接地线及电源线上加装滤波器。

4) 采用低辐射设备

低辐射设备是指在设计和生产计算机设备时,就对可能产生电磁辐射的元器件、集成电路、连接线、显示器等采取了防辐射措施,把电磁辐射抑制到最低限度。由于制造低辐射设备所使用的材料成本较高,它的造价也比较昂贵。使用低辐射计算机设备是防止计算机电

磁辐射泄密的较为根本的防护措施。

5）隔离和合理布局

隔离是将信息系统中需要重点防护的设备从系统中分离出来，加以特别防护，并切断其与系统中其他设备间的电磁泄漏通路。合理布局是指合理放置信息系统中的关键设备，并尽量拉大涉密设备与非安全区域的距离。让计算机房远离可能被侦测的地点，这是因为计算机辐射的距离有一定限制，超过 300m，即使攻击者接收到辐射信号也很难还原。对于一个单位而言，计算机房尽量建在单位辖区的中央地区而不是边缘地区。若一个单位辖区的半径小于 300m，距离防护的效果就有限。

对计算机与外部设备究竟要采取哪些防泄漏措施，要根据计算机中的信息的重要程度而定。对于企业而言，需要考虑这些信息的经济效益，在选择保密措施时，不应该花费 100 万元去保护价值 10 万元的信息，对于军队则需要考虑这些信息的保密级别。

3. 防电磁干扰

防止计算机受到电磁干扰的主要手段有接地、屏蔽、滤波。

1）接地

良好的接地系统，一是可以消除各电路之间流经公共阻抗时所产生的公共抗阻干扰和雷击，避免计算机电路受磁场和电位差的影响，二是可保证设备及人身安全。理想的接地面为零电位，各接地点之间无电位差。在接地设计时，应注意交流地、直流低、防雷地和安全地的接地线要分开，不要互连，复杂电路要采用多点接地和公共地等。

2）屏蔽

电磁屏蔽是对两个空间区域之间进行金属的隔离，以控制电场、磁场和电磁波由一个区域到另一个区域的感应和辐射。具体讲，就是用屏蔽体将元部件、电路、电缆或整个系统的干扰源包围起来，防止干扰电磁场向外扩散；用屏蔽体将接收电路、设备或系统包围起来，防止它们受到外界电磁场的影响。在计算机工程中，凡是收到电磁场干扰的地方都可以用屏蔽的方法来削弱干扰，以确保计算机正常运行。

屏蔽材料通常采用高导电性的材料，如铜板、铜箔、铝板、铝箔、钢板或金属镀层、导电涂层。由于电磁干扰无孔不入，因此屏蔽体上应尽量少留开口，还可以根据需要采取不同的金属材料组成的多层屏蔽体。

3）滤波

滤波是抑制和防止干扰的一项重要措施。在一定的频带内，滤波器衰减很小，电能很容易通过，而在此频带外衰减则很大，能有效抑制传输。应在计算机的线路板上采取适当的滤波措施，防止外部的电磁干扰，同时又能使脉冲信号的高频成分大大减少，从而使线路板的辐射得到改善。

3.4 媒体（介质）安全

3.4.1 媒体安全面临的威胁

常见的信息存储媒体有磁盘、光盘、U 盘、移动硬盘、打印纸等，这些媒体上存储了大量有用信息甚至机密信息，是各类黑客或攻击者进行盗窃、破坏和篡改的目标。光盘、U 盘、

移动硬盘、打印纸等体积小、易携带，成为最容易造成信息泄露的设备，下面主要讨论硬盘面临的安全威胁和安全防护措施。

硬盘面临的安全威胁有：(1)目前 PC 的硬盘是很容易安装和拆卸的，导致硬盘容易被盗。(2)硬盘上的文件几乎没有任何保密措施，如果硬盘被盗了，那么硬盘上的文件、信息、办公秘密、商业机密也就暴露无疑。目前我们比较常用的办公软件 Word 可以通过设置保护密码和限制访问进行文件保护，但为了办公方便，使用得极少，而且在强大的破译软件面前，这种密码保护根本不堪一击。(3)文件删除操作留下的隐患。文件的删除操作是人们经常执行的操作，系统在执行删除操作时，数据是否在磁盘上不存在了呢？实际上，文件删除操作仅仅在文件目录中作了一个标记，并没有删除文件本身数据存储区，数据仍然残留在磁盘上，直到新的数据覆盖，如图 3.4 所示。这段时间内信息泄露的可能性比较大。(4)硬盘本身的脆弱性。磁盘本身很容易被划坏，或被各种硬物碰伤或受潮霉变，硬盘上的数据也随之而变得无法读取。

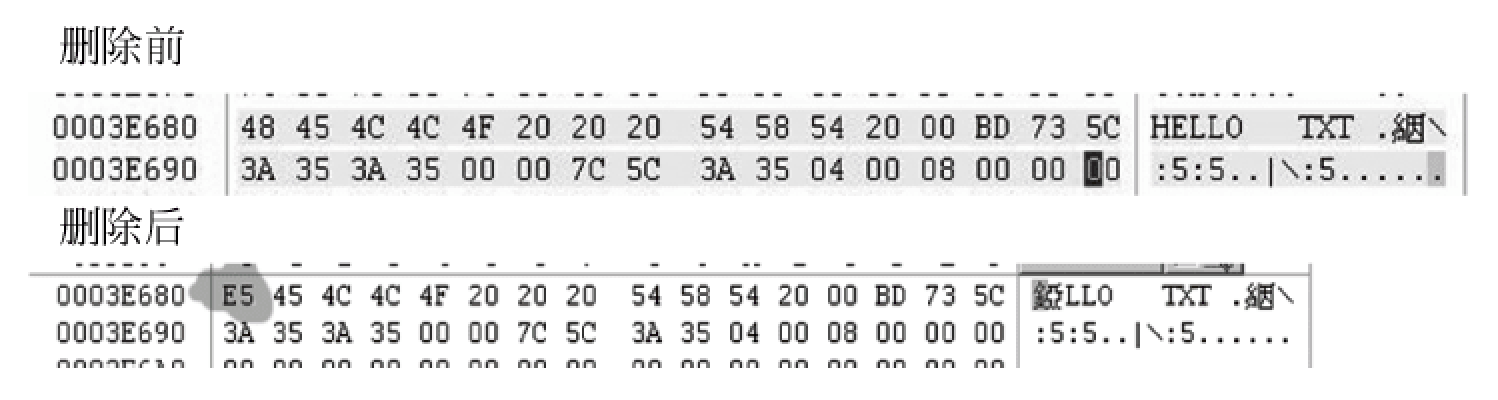


图 3.4　文件删除前后磁盘存储区的变化

媒体面临的安全威胁还来自于管理方面的缺陷，在媒体的使用和管理上存在如下 4 个方面的缺陷：

(1) 缺乏对媒体的管理和维护能力，一旦存储有重要、敏感信息的媒体发生故障，只能销毁或冒着泄密的重大危险到固定维修点甚至去国外维修。

(2) 对存储有敏感信息的媒体没有专门的存放场所，而是和一般办公文件一起存放，造成一旦办公场所发生火灾等突发危险，媒体随之而遭殃。

(3) 缺乏对媒体的分类和拷贝限制。没有根据媒体的重要程度进行分类存放，且对媒体中信息的拷贝流程没有严加管理，信息拷贝几乎人手一份，所以媒体中的信息也毫无秘密可言。

(4) 缺乏媒体的管理办法。没有形成对媒体的分类存放和拷贝管理方法，以及相应的拷贝登记制度、媒体处理和销毁制度。

3.4.2　媒体安全防护

媒体的安全防护应从加强磁盘安全保密控制和加强媒体安全管理两方面入手。

1. 加强磁盘安全保密控制

可以通过磁盘加密技术和磁盘信息清除技术加强对磁盘及其存储信息的安全保护。

磁盘加密技术是指使用加密工具对存储在磁盘上的信息进行加密，即使存储信息被第三方窃取或复制，也很难读懂，从而保证信息不被泄露。具体的磁盘信息加密技术还可细分为文件加密、目录加密、数据库加密和整盘数据加密。具体应用可视磁盘信息的保密强度要求而定。

磁盘清除技术可以分为直流消磁法和交流消磁法两种。直流消磁法是使用直流磁头将磁盘上原先记录信息的剩余磁通全部以一种形式的恒定值来代替。通常，完全格式化方式格式化磁盘就是这种方法。交流消磁法是使用交流磁头将磁盘上原先所记录信息的剩余磁通变得极小，这种方法的消磁效果比直流消磁法的效果要好，消磁后磁盘上的残留信息强度可比消磁前下降 90dB。

2. 加强媒体安全管理

(1) 设置管理人员对媒体进行专门管理。一方面所有对存储媒体的访问应当由管理员统一进行管理，另一方面管理员要负责所有媒体的接收和发出，并做好相应的核准和记录工作。

(2) 做好媒体的归档工作。任何媒体都要有完整的归档记录，归档文件要清楚、齐全，一旦投入使用，任何人未经批准不得增、删、改。

(3) 加强敏感媒体的管理。所有媒体应采用物理方法标识出密级；造册登记，编制目录，集中管理；复制、传递、使用、发放都要有审批签字手续，归还时要严格复核手续等；凡属规定密级的各种记录媒体，禁止使用中途转借给他人；保密的存储介质或文件在不使用时应存放在安全的地点并锁在安全器内；销毁必须登记，并由承办人填写销毁记录。

(4) 当存储媒体不使用时，在转交给他人使用之前，不能只作简单的删除操作，而必须把存储在上面的保密数据彻底格式化。

(5) 如果要对关键敏感性的媒体进行销毁，可以采取物理粉碎、强磁场消磁和高温焚烧等方法进行销毁，同时也要注意销毁的登记。

3.5 系统安全和可靠性技术

1. 系统安全和可靠性的定义

系统安全是指为保证信息系统安全可靠运行而采取的安全措施，可用性和可靠性是衡量系统安全的主要指标。

可用性是指系统在规定条件下，完成规定功能的能力。系统可用性用可用度来衡量。系统在 t 时刻处于正确状态的概率称为可用度，用 $A(t)$来表示。

$$A(t)=\text{平均无故障时间}/(\text{平均无故障时间}+\text{平均修复时间})$$

平均无故障时间(Meantime Between Failures，MTBF)是指两次故障之间能正常工作的平均时间。故障是指由于部件的物理失效、环境应力的作用、操作错误或不正确的设计，引起系统的硬件或软件的错误状态。故障既可能是元件故障、软件故障，也可能是人为攻击造成的系统故障。

平均修复时间(Meantime Repair a Failure，MTRF)是指从故障发生到系统恢复所需要的平均时间。

可用性还表现在以下三个方面：

1) 可靠性

如果系统从来没有故障，那么可用性就是 100%，但这基本上是不可能的，所以引进一

个辅助参数——可靠性(Reliability),即在一定的条件下,在指定的时期内系统无故障地执行指定任务的可能性。系统可靠性采用可靠度来衡量。可靠度是指在 t_0 时刻系统正常运行的条件下,在给定的时间间隔内,系统仍然能执行其功能的概率。

2) 可维修性

可维修性是指系统发生故障时容易进行修复以及平时易于维护的程度。可维修性可表现为平均修复时间(MTRF),在指定时间内恢复服务的可能性。

3) 维修保障

维修保障即系统发生故障时,后勤支援的能力。

因计算机系统硬、软件故障降低信息系统的可靠性,提高信息系统可靠性一般采取避错和容错技术,为抵御灾难造成的信息系统不可用,可采用容灾备份技术实现对灾难的容忍。

2. 提高信息系统可靠性的措施

提高信息系统的可靠性一般采取避错、容错和容灾备份技术。

1) 避错

避错即通过提高信息系统软硬件的质量以抵御故障的发生,要求组成系统的各个部件、器件、软件具有高可靠性,不允许出错或出错率极低。通过精选元器件、严格的工艺、精心的设计来提高可靠性。在现有条件下避错设计是提高系统可靠性的有效办法。受人们认知的局限性和技术水平的限制,避错不能完全消除错误的发生。

2) 容错

一个系统无论采用多少避错方法,对于可靠性的提高都是有限的,因为不可能保证永远不出错。因此,还要发挥容错技术,使得在故障发生时,系统仍能继续运行,提供服务与资源。容错设计是在承认故障的情况下进行的,是指在计算机内部出现故障的情况下,计算机仍能正确地运行程序并给出正确结果的设计。

3) 容灾备份

容灾备份是信息系统安全的基础设施,对重要信息系统建立容灾备份系统,可以防范和抵御灾难发生给信息系统造成的毁灭性打击。

3.6　容错技术

容错是一种可靠性保障技术,利用冗余的资源使计算机具有容忍故障的能力,即在发生故障的情况下,计算机仍有能力完成指定的任务或继续向外提供正确的服务。

人们对容错技术的研究开始得很早,1952 年,冯·诺依曼就在美国加利福尼亚理工学院做过 5 个关于容错理论研究的报告,他的精辟论述成为以后容错研究的基础。容错技术最早在硬件上研究和实现,在 1950 年至 1970 年得到了重大的发展,并成为一种成熟的技术应用于实际系统中,如双 CPU、双电源。到 20 世纪 60 年代末,出现了以自检、自修计算机 STAR 为代表的容错计算机。20 世纪 70 年代容错技术的应用和研究范围迅速扩大至交通管制、工厂自动化、电话开关等领域,并且出现了用软件实现容错的 SIFT 计算机。20 世纪 80 年代,容错技术的研究随着计算机的普及深入到各个行业,许多公司生产的容错计算机如 Stratus 容错计算机系列,IBM System88 等已商品化并推入市场。

容错技术主要是通过冗余设计来实现的,所谓冗余就是超过系统实现正常功能的额外资源。它以增加资源的办法来换取可靠性。根据增加资源的不同,容错技术可以分为硬件容错、软件容错、数据容错和时间容错。

1. 硬件容错

硬件容错用以避免由于硬件造成的系统失效,通过硬件的物理备份来获得容错能力,如冗余处理器、冗余内存、冗余电源等。广泛应用的硬件冗余之一是硬件堆积冗余,在物理级通过原件的重复获得。硬件容错还可以通过待命储备冗余实现,系统中设置 $m+1$ 个模块,只有一个处于工作状态,其余 m 块都处于待命接替状态,一旦工作模块出了故障,立刻切换到一个待命储备模块,当换上的储备模块发生故障时,又切换到另一储备模块,直至资源枯竭。目前,硬件容错广泛应用于信息关键系统中,例如民航飞机中总有几套计算机系统同时运行,磁盘冗余阵列是硬件容错的典型。

2. 软件容错

软件容错用于避免由于软件引起的系统失效。软件容错的基本思想是用多个不同的软件执行同一功能,利用软件设计差异来实现容错。通过提供足够的冗余信息与算法程序,使系统在实际运行中能够及时发现程序错误,采取补救措施,保证整个计算的正确运行。执行同一任务采用的不同软件程序组成一个有机整体,完成错误检测,程序系统重组及系统恢复等多项功能,达到利用设计差异实现容错的目的。

3. 数据容错

数据容错是指增加额外的数据位以检测或纠正数据在运算、存储及传输中的错误。编码技术是一种数据容错技术,它通过在数据中附加冗余的信息以达到故障检测和故障掩蔽或容错的目的,包括检错编码与纠错编码技术。检错编码可以自动发现错误,而纠错编码具有自动发现错误和纠正错误的能力。编码技术常用在信息的存储、传输和处理中。在计算机系统中,常用的编码技术有奇偶校验码、循环冗余校验码和扩展海明码等。

4. 时间容错

时间容错是通过消耗时间资源来实现容错,其基本思想是重复执行指令或程序来消除故障带来的影响。按照重复运算在指令级还是程序级可以分为指令复执和程序卷回。指令复执就是当机器检测到错误后,让当前指令重复执行若干次,如果错误是瞬时的,在指令复执期间,有可能不再出现,程序就可继续向前运行,如果在指令复执期间不能纠正错误,则需要通过人工干预或调用诊断程序来消除错误。程序卷回是重复执行一小段程序,常用回滚技术实现。例如将机器运行的某一时刻作为检查点,此时检查系统运行的状态是否正确,无论正确与否,都将这一状态存储起来,一旦出现运行故障,就返回到最近一次正确的检查点重新运行。

这 4 种容错技术中最重要,也是应用最多的是硬件容错和软件容错,也是本节学习的重点。

3.6.1 硬件容错

硬件容错是通过硬件冗余实现的,硬件冗余通过在一个硬件部件中提供两个或多个物理实体实现冗余,是在给定器件可靠性的前提下提高系统组成部件可靠性的有效方法。硬

件冗余有三种基本形式。

(1) 被动冗余：在无须其他操作的情况下，通过屏蔽故障实现容错。三模冗余(Triple Modular Redundancy,TMR)是被动冗余的典型代表。TMR 首先利用三份硬件同时进行相同的功能的计算，再通过投票器从三份计算结果中选择正确的计算结果。如果三份硬件中有一个发生故障，产生错误计算结果，则投票器将剩余两个硬件产生的相同计算结果作为最终计算结果。根据应用的不同，三模冗余的硬件可以是处理器、存储器、电源等。值得注意的是，由于投票器使用少数服从多数的算法，因此 TMR 仅能够屏蔽一个故障部件，为了屏蔽更多的故障部件，则需要使用 5MR、7MR 等更高模的冗余。

(2) 主动冗余：在容错之前首先进行故障检测，在故障检测后再进行故障定位和故障恢复工作，从而移除系统中的故障部件。备用备件(Standby Sparing)是一种主动冗余方法，在一个 n 模冗余的备用备件方法中，n 个模块中只有一个是活跃的，而其他 $n-1$ 个模块都作为备份使用。每个模块都有一个故障检测器，并将所有模块连接在一个选择器上。当活跃模块的故障检测器发现故障后，选择器将从备份模块中选择一个模块作为新的活跃模块。

(3) 混合冗余：结合了被动冗余和主动冗余的方法，利用主动冗余防止大量错误的产生，利用被动容错实现故障部件的更换。例如我们可以在使用 TMR 的基础上，将投票器选择的正确结果反馈给所有冗余模块，各个模块通过将自己的计算结果与反馈结果进行比较，从而判断该模块是否为故障模块，并决定是否将自己移除。

3.6.2　软件容错

随着软件功能和性能的飞速提升，软件变得越来越复杂，软件由于设计缺陷或误操作而引发系统错误的概率也不断提高。统计数据表明，当前计算机系统中 60%～90%的故障是由软件故障引起的。由于软件不像硬件那样存在制造缺陷，也不会产生磨损，所以软件故障大多是由设计故障引起的。

软件容错用于提高软件系统的可靠性，通过提供足够的冗余信息和算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高软件可靠性。软件容错使软件在出错时仍能向外提供正常或降级服务，避免出现重大人身或财产损失。软件容错技术的方法主要有 N 版本程序设计和恢复块方法。

1. 恢复块方法

1975 年，B. Randell 提供了一种动态故障屏蔽技术——恢复块方法，如图 3.5 所示，这也是最早的一种软件容错技术。恢复块通常与判决器一起使用。在使用恢复块的系统中，系统被划分成一个个故障恢复块，整个系统由这些故障恢复块组成。每个块包含一个主块和一些用来替换的后备模块。主块在第一时间运行，其输出要通过判决器来检查可接受性。这是整个设计中的瓶颈，因为判决器不知道正确的输出该是什么。判决器执行检查，检查输出是否在某个可接受的范围内，或输出有没有超出允许的最大变化率。例如，如果任务是计算一艘船的位置，如果前后几微秒之间的距离差有 1000 海里，这样的结果显然是不正

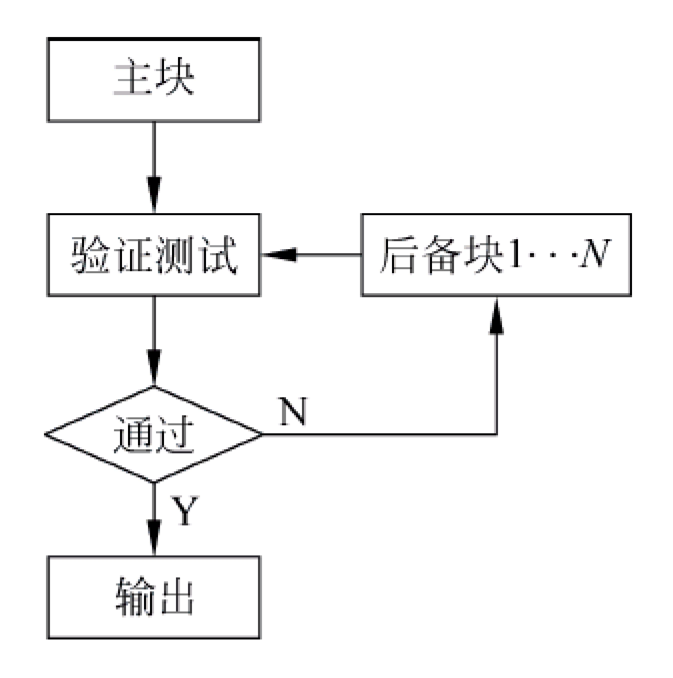


图 3.5　恢复块方法

确的。

主块得到输出后，立即进行接受测试，当接受测试判断输出不可接受时，系统将回滚并恢复到主块运行之前的状态，然后调用第二个模块，运行获得结果并进行接受测试。如果调用也失败的话，则继续调用另外的替换模块，系统重复这样的操作，直到用完所有模块，或超出规定的时间限制。

使用恢复块方法会引起时间开销，当首要执行模块失败时就发生了时间开销，包括保存全局状态和启动一个或多个替换的模块。这使得恢复块系统很复杂，因为在重试下一个之前，需要系统状态具有回滚的能力。当然也可以通过其他方法完成回滚，例如使用硬件支持该操作。

设置接受测试时，设计人员常常面临一些难题，如果允许的范围太严格，那么接受测试将产生大量的错误警报。如果设置太宽松的话，把错误的输出当作正确结果接受的可能性会大大增大。因此设置接受测试时必须从实际出发，根据需求进行相应的设置工作。

2. *N* 版本技术（NVP）

N 版本技术是一种静态的故障屏蔽技术，其设计思想是 *N* 个独立生成的功能相同的程序同时执行，使用表决器比较各个版本产生的结果，并将其中一个作为正确的结果给出。这种容错方法依赖于不同版本程序之间的独立性。该技术已经运用到很多实际系统中，例如铁路交通控制系统、飞行控制系统。

在 *N* 版本软件系统中，有 *N* 个不同的模块同时独立地执行。每个模块以不同的方式完成相同的任务，各自向表决器提交它们的结果，由表决器确定正确的结果，并作为模块的结果返回。利用设计多样性得到的 *N* 版本软件系统能克服大多数软件中出现的设计故障。*N* 版本软件的一个重要特性就是系统包含多个版本软件和多种类型的硬件，目的是通过增加差异以避开共有的故障。开发 *N* 版本软件过程中，对于每个不同版本，尽可能以不同的方式实现，包括不同的工具（例如静态和动态的分析器，辅助调试的专家系统等工具）、不同的编程语言以及不同的环境。每个开发小组在编程期间也要尽可能地减少交流。只有满足设计的多样性，*N* 版本软件才能真正做到容错。

恢复块和 *N* 版本软件之间的不同之处并不多，但却非常突出。传统的恢复块方法中，用来替换的块逐个执行，直到判决器找到可接受的结果，*N* 版本软件方法通常在 *N* 份冗余的硬件上同时执行这些不同版本的软件。在逐个重试的过程中，尝试多个替换版本的时间开销可能很大，这种方法尤其不适用于实时系统，相反的，同时运行的 *N* 版本软件系统需要 *N* 份冗余硬件和通信网络连接它们。这两种方法的另一个重要不同在于判决器和表决器。恢复块方法需要为每个模块建立一个特定的判决器，而 *N* 版本软件方法中，只需要使用一个简单的表决器即可。在实际开发中，设计人员要全面衡量它们的优缺点，并结合应用的实际需求，进行折中考虑，尤其是性能和资金的开销方面，从而确定哪种方案更适合工程。

为了体现对设计的冗余，NVP 系统的 *N* 份程序必须采取不同的方法或由不同的人独立设计。NVP 系统的结构如图 3.6 所示。

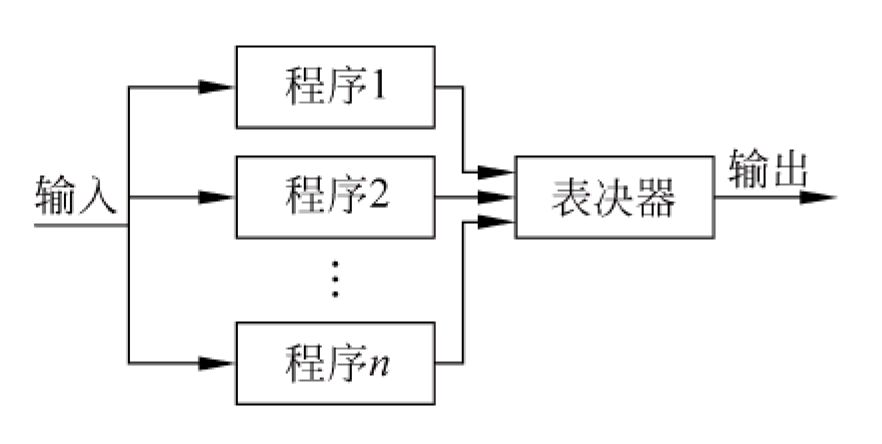


图 3.6　NVP 系统结构

3.7 信息系统灾难恢复技术

随着信息化的发展,越来越多关键数据和业务集中到信息系统中,社会对信息系统的依赖性越来越强,而信息系统易受到地震、火灾、人为误操作、硬件故障等诸多侵扰,一方面数据丢失和损坏将造成难以估量的损失,另一方面即使短时间的系统停机也将造成业务停顿和经济损失,因此灾难备份和恢复成为迫切需要解决的问题,重要信息系统必须建立容灾备份系统,以防范和抵御灾难带来的打击。美国国防部提出的信息保障模型 PDRR 中就包含了恢复环节,灾难恢复是信息系统安全的重要组成部分。

3.7.1 概述

传统的数据备份技术和服务器集群技术足以避免由于各种软硬件故障、人为操作失误和病毒侵袭所造成的破坏,保障数据安全,但是当面临大范围灾害性突发事件,如地震、火灾、恐怖袭击时,上述技术就无能为力了。此时若想迅速恢复应用系统的数据,保持企业的正常运行,就必须建立异地的灾难备份系统(即容灾系统)。美国 Minnesota 大学的研究表明,遭遇灾难的同时又没有灾难恢复计划的企业,超过 60%以上的在 2~3 年后将退出市场。在美国“9·11”事件中,很多公司多年积累的经营数据毁于一旦,公司处于崩溃的边缘,而一些建立了容灾系统的公司,如总部设在世贸中心的摩根-斯坦利公司,在第二天就恢复了正常运转。这一事例再次唤起人们对容灾技术的重视。

业务连续性和灾难恢复起步于 20 世纪 70 年代中期的美国,历史性标志是 1979 年在美国宾夕法尼亚州的费城建立了专业商业化的容灾备份中心并对外提供服务。20 世纪 90 年代后期,千年虫问题促进了业务连续性和灾难恢复管理的进一步深入和发展。2001 年轰动一时的“9·11”恐怖袭击事件不仅造成了重大的人员伤亡和财产损失,一批设在世贸中心的公司因为重要数据的毁灭而再也无法正常营业。“9·11”给大家带来的深刻的启示就是容灾备份是信息系统安全的重要设施,重要信息系统必须构建容灾备份系统,以防范和抵御灾难带来的毁灭性打击。

我国业务连续性和灾难恢复工作起步于 20 世纪 90 年代末,这时一些单位在信息化建设的同时,开始关注数据安全的保护,开展了数据备份工作。随后,千年虫问题和“9·11”事件也极大触动了我国灾难恢复管理的发展和成熟。

2003 年,中共中央办公厅、国务院办公厅下发了《国家信息化领导小组关于加强信息安全保障工作的意见》,在文件中要求要高度重视灾难备份工作。为贯彻落实中央指示,国务院信息化工作办公室于 2004 年 9 月下发了《关于做好重要信息系统灾难备份工作的通知》,文件强调了“统筹规划、资源共享、平战结合”的灾难恢复工作原则。为进一步推动 8 个重点行业(银行、证券、保险、电力、民航、铁路、海关、税务)加快实施灾难恢复工作,国务院信息化工作办公室于 2005 年 4 月下发了《重要信息系统灾难恢复指南》,文件指明了灾难恢复的流程,容灾备份中心的等级划分及灾难恢复预案的制定,使得灾难恢复建设迈上了一个新的台阶。2007 年在《重要信息系统灾难恢复指南》基础上,编制并正式发布了国家标准 GB/T 20988—2007《信息安全技术信息系统灾难恢复规范》来指导信息技术容灾备份系统的建设。

3.7.2 灾难恢复的级别和指标

1. 灾难恢复的定义

在《重要信息系统灾难恢复规划指南》中对灾难有明确定义：灾难是由于人或自然原因造成的信息系统运行严重故障或瘫痪，使信息系统支持的业务功能停顿或服务水平不可接受、达到特定时间的突发性事件，通常导致信息系统需要切换到备用场地运行。灾难主要包括地震、火灾、水灾、战争、恐怖袭击、设备系统故障、人为破坏等无法预料的突发事件。

灾难恢复是指利用技术、管理手段及相关资源确保关键数据、关键数据处理信息系统、关键业务在灾难发生后可以恢复和重续运营的过程。灾难恢复的最高目标是实现数据零丢失和业务连续性。

2. 容灾备份系统的种类

按照建立容灾系统目标的不同，容灾备份系统可以分为两种，即数据容灾、应用容灾。

数据容灾是最常见的容灾备份方式，是指建立一个异地的备份数据系统，该系统是对本地系统关键应用数据的实时复制，也可比本地数据略微滞后。数据容灾的主要目的是保证企业关键数据的完整性和可用性。在数据容灾这个级别，发生灾难时应用会中断，服务器必须暂停业务来进行异地恢复，这种方式的优点是成本低、构建简单。但是对需要保持7×24小时连续服务的企业来说，数据级容灾方式显然是不够的。

应用级容灾是在数据容灾的基础上，同时将应用程序的处理状态进行备份，其实现方式是在异地建立一套完整的、与本地数据系统相当的备份应用系统（可以同本地应用系统互为备份，也可与本地应用系统共同工作）。当灾难发生时，异地的应用容灾中心可以接替原来的系统继续工作，保持业务的连续性。应用容灾是更高层次的容灾系统。

3. 灾难备份系统的级别

设计一个容灾备份系统需要考虑多方面的因素，包括备份/恢复的数据量大小、应用数据中心和备援数据中心之间的距离和连接方法、灾难发生时所要求的恢复速度、备援中心的管理和经营方法，以及可投入的资金多少等。根据这些因素可将容灾备份系统划分为不同的级别，分别适用于不同的规模和应用场合。

1）国际上的灾难恢复等级划分

灾难恢复的国际标准是1992年Anaheim提出的SHARE 78，将灾难恢复由高到低划分为7级：

（1）第0级：没有异地数据（No Off-site Data）。

第0级没有任何异地备份或应急计划，数据仅在本地进行备份恢复，没有送往异地。事实上这一层并不具备真正灾难恢复的能力。

（2）第1级：卡车运送访问方式（Pickup Truck Access Method，PTAM）。

第1级要求必须设计一个灾难恢复应急方案，能够备份所需要的信息并将它保存在异地，灾难恢复时将根据需要，有选择地搭建备援的硬件平台并在其上恢复数据。PTAM指将本地备份的数据用交通工具送到远方。这种方案相对来说成本较低，但难于管理。

PTAM是一种广泛使用的容灾系统，备份数据被送往远离本地的异地保存，可抵御大

规模的灾难事件。灾难发生后，需要按预定的数据恢复方案购置和安装备援硬件平台、恢复系统和企业数据，并重新与网络连接。这种容灾方案成本低(仅需要传输工具和存储设备的消耗)，且易于配置。但当数据容量增大时，备份数据难以管理，用户难以及时知道所需的数据存储在什么地方。

当备援系统开始工作后，首先应及时恢复关键应用，非关键应用可根据需要慢慢恢复，因为 PTAM 的备份地点事先往往只有很少的硬件设备，因此我们将其称为冷备份站点，它的恢复时间往往较长，如一星期甚至更久。

(3) 第 2 级：PTAM＋热备份中心(PTAM＋Hot Center)。

第 2 级在第 1 级的基础上再加上热备份中心以进一步灾难恢复。热备份中心拥有足够的硬件和网络设备，当主数据中心破坏时可切换用于支持关键应用的备援站点。对于十分关键的应用，必须由热备份站点在异地提供支持。这样当灾难发生时才能及时恢复。在第 2 级容灾系统中，平时备份数据用 PTAM 的方法存入备份数据仓库，当灾难发生的时候，备份数据再被运送到一个热备份站点。虽然移动数据到一个热备份站点增加了成本，但却缩减了灾难恢复的时间，一般在一天左右。

(4) 第 3 级：电子链接(Electronic Vaulting)。

第 3 级是在第 2 级的基础上用电子链路取代卡车进行备份数据传送的容灾系统，热备份站点和主数据中心在地理上必须远离，备份数据通过网络传输。由于热备份站点要持续运行，因此系统成本高于第 2 级，但进一步提高了灾难恢复的速度，典型的恢复时间在一天以内。

(5) 第 4 级：活动状态的备份中心(Active Secondary Site)。

第 4 级要求地理上分开的两个站点同时处于工作状态并相互管理彼此的备份数据，另一项重大的改进就是两个站点之间可以相互分担工作负载，站点一可以成为站点二的备份；反之亦然，备援行动可以在任何一个方向发生。关键的在线数据不停地在两个站点之间复制和传送着，灾难发生时，另一站点可通过网络迅速切换用于支持关键应用。但是该系统自最近一次数据复制以来的业务数据将会丢失，其他非关键应用也将需要手工恢复。第 4 级容灾系统把关键应用的灾难恢复时间降低到了小时级或分钟级。

(6) 第 5 级：两个活动的数据中心，两步提交(Two-Site，Two-Phase Commit)。

第 5 级与第 4 级的结构类似，在满足第 4 级所有功能要求的基础上，进一步提供了两个站点间的数据互作镜像(数据库的一次提交过程会同时更新本地和远程数据库中的数据)。数据库的两步提交方法保证了任何一项事务在被接收以前，两个站点间的数据都必须同时被更新。在备援站点中需要配备一些专用硬件设备，以保证在两个站点之间自动分担工作负载和两步提交的正确执行，因为采用了两步提交来同步数据，在两个站点间互作镜像，所以当灾难发生时，仅仅只有传送中尚未完成提交的数据被丢失，恢复的时间被降低到了分钟级。

(7) 第 6 级：0 数据丢失(Zero Data Loss)。

第 6 级是灾难恢复的最高级别，可以实现零数据丢失。只要用户按下 Enter 键向系统提交了数据，那么不管发生了什么灾难性事件，系统都能保证该数据的安全。所有的数据都将在本地和远程数据库之间同步更新，当发生灾难事件时，备援站点能通过网络侦测故障并立即自动切换，负担起关键应用。第 6 级是容灾系统中最昂贵的方式，但也是速度最快的恢

复方式。

第4级、第5级和第6级容灾系统具有类似的系统框架结构，区别在于数据备份管理软件的差异和备援站点内硬件配置的不同，进而导致了系统成本和性能的差异。第4级的容灾系统只需要配置远程系统备份软件即可工作；第5级容灾系统依赖于数据库系统的两步提交来保持数据的同步；第6级容灾系统则需要配置复杂的数据管理软件和专用的硬件设备，以保证灾难发生时的零数据丢失和备援站点的即时切换。

2）我国灾难恢复等级划分

在GB/T 20988-2007《信息安全技术信息系统灾难恢复规范》中，根据支持灾难恢复各个等级所需要的资源，即数据备份系统、备用数据处理系统、备用网络系统、备用基础设施、技术支持能力、运行维护管理能力和灾难恢复预案7个要素划分了6个灾难恢复等级。

第一级：基本支持；

第二级：备用场地支持；

第三级：电子传输和部分设备支持；

第四级：电子传输及完整设备支持；

第五级：实时数据传输及完整设备支持；

第六级：数据零丢失和远程集群支持。

(1) 第一级：基本支持。

在第一级中，每周至少做一次完全数据备份，并且备份介质场外存放，同时还需要有符合介质存放的场地；单位要制定介质存放、验证和转储的管理制度，并按介质特征对备份数据进行定期的有效性验证；单位需要指定经过完整测试和演练的灾难恢复预案，具体技术和管理支持如表3.2所示。

表3.2 灾难恢复第一级要求

	要 素	要 求
A.1.1	数据备份系统	(1) 完全数据备份至少每周一次 (2) 备份介质场外存放
A.1.2	备用数据处理系统	—
A.1.3	备用网络系统	—
A.1.4	备用基础设施	有符合介质存放条件的场地
A.1.5	技术支持	—
A.1.6	运行维护支持	(1) 有介质存取、验证和转储管理制度 (2) 按介质特征对备份数据进行定期的有效性验证
A.1.7	灾难恢复预案	有相应的经过完整测试和演练的灾难恢复预案

(2) 第二级：备用场地支持。

第二级相当于在第一级的基础上，增加了在预定时间内能调配所需使用的数据处理设备、通信线路和网络设备到场要求；并且需要有备用的场地，它能满足信息系统和关键功能恢复运行的要求；对于单位的运维能力，也增加了具有备份场地管理制度和签署符合灾难恢复时间要求的紧急供货协议，具体技术和管理支持如表3.3所示。

表 3.3　灾难恢复第二级要求

	要　素	要　求
A.2.1	数据备份系统	(1) 完全数据备份至少每周一次 (2) 备份介质场外存放
A.2.2	备用数据处理系统	灾难发生时能在预定时间内调配所需的数据处理设备
A.2.3	备用网络系统	灾难发生时能在预定时间内调配所需的通信线路和网络设备
A.2.4	备用基础设施	(1) 有符合介质存放条件的场地 (2) 有满足信息系统和关键业务恢复运作要求的备用场地
A.2.5	技术支持	—
A.2.6	运行维护支持	(1) 有介质存取、验证和转储管理制度 (2) 按介质特征对备份数据进行定期的有效性验证 (3) 具有备份场地管理制度 (4) 与相关厂商签署符合灾难恢复时间要求的紧急供货协议 (5) 与相关厂商签署符合灾难恢复时间要求的备用通信线路协议
A.2.7	灾难恢复预案	有相应的经过完整性测试和演练的灾难恢复预案

(3) 第三级：电子传输和部分设备支持。

第三级要求配置部分数据处理设备、部分通信线路和网络设备；要求每天实现多次的数据电子传输，并在备用场地配置专职的运行管理人员；对于运行维护支持而言，要求具备备用计算机处理设备维护管理制度和电子传输备份系统运行管理制度，具体技术和管理支持如表 3.4 所示。

表 3.4　灾难恢复第三级要求

	要　素	要　求
A.3.1	数据备份系统	(1) 完全数据备份至少每天一次 (2) 备份介质场外存放 (3) 每天多次利用通信网络将关键数据定时批量传送至备用场地
A.3.2	备用数据处理系统	配置灾难恢复所需的部分数据处理设备
A.3.3	备用网络系统	配备部分通信线路和网络设备
A.3.4	备用基础设施	(1) 有符合介质存放条件的场地 (2) 有满足信息系统和关键业务恢复运作要求的备用场地
A.3.5	技术支持	在备用场地有专职的计算机机房运行管理人员
A.3.6	运行维护支持	(1) 按介质特征对备份数据进行定期的有效性验证 (2) 有介质存取、验证和转储管理制度 (3) 有备用计算机机房管理制度 (4) 有备用数据处理设备硬件维护管理制度 (5) 有电子传输数据备份系统运行管理制度
A.3.7	灾难恢复预案	有相应的经过完整性测试和演练的灾难恢复预案

(4) 第四级：电子传输和完整设备支持。

第四级相对于第三级中的部分数据处理设备和网络设备而言，需配置灾难恢复所需的全部数据处理设备、通信线路和网络设备，并处于就绪状态；备用场地也提出了 7×24 小时运行的要求，同时，对技术支持人员和运维管理要求也有相应的提高，具体技术和管理支持如表 3.5 所示。

表 3.5 灾难恢复第四级要求

	要　素	要　求
A.4.1	数据备份系统	(1) 完全数据备份至少每天一次 (2) 备份介质场外存放 (3) 每天多次利用通信网络将关键数据定时批量传送至备用场地
A.4.2	备用数据处理系统	配置灾难恢复所需的全部数据处理设备,并处于就绪状态或运行状态
A.4.3	备用网络系统	配备灾难恢复所需的通信线路和网络设备,并处于就绪状态
A.4.4	备用基础设施	(1) 有符合介质存放条件的备用场地 (2) 有符合备用数据处理系统和备用网络设备运行要求的场地 (3) 有满足关键业务功能恢复运作要求的场地 (4) 以上场地应保持 7×24 运作
A.4.5	技术支持	在备用场地有: (1) 7×24 专职计算机机房管理人员 (2) 专职数据备份技术支持人员 (3) 专职硬件、网络技术支持人员
A.4.6	运行维护支持	(1) 按介质特征对备份数据进行定期的有效性验证 (2) 有介质存取、验证和转储管理制度 (3) 有备用计算机机房运行管理制度 (4) 有硬件和网络运行管理制度 (5) 有电子传输数据备份系统运行管理制度
A.4.7	灾难恢复预案	有相应的经过完整性测试和演练的灾难恢复预案

(5) 第五级:实时数据传输和完整设备支持。

第五级相对于第四级的数据电子传输而言,要求采用远程数据复制技术,利用网络将关键数据实时复制到备用场地;备用网络应具备自动或集中切换能力;备用场地有 7×24 小时专职数据备份、硬件、网络技术支持人员,具备较严格的运行管理制度,具体技术和管理支持如表 3.6 所示。

表 3.6 灾难恢复第五级要求

	要　素	要　求
A.5.1	数据备份系统	(1) 完全数据备份至少每天一次 (2) 备份介质场外存放 (3) 用远程数据复制技术,并利用通信网络将关键数据实时复制到备份场地
A.5.2	备用数据处理系统	配置灾难恢复所需的全部数据处理设备,并处于就绪状态或运行状态
A.5.3	备用网络系统	(1) 配备灾难恢复所需的通信线路和网络设备,并处于就绪状态 (2) 具备通信网络自动或集中切换能力
A.5.4	备用基础设施	(1) 有符合介质存放条件的备用场地 (2) 有符合备用数据处理系统和备用网络设备运行要求的场地 (3) 有满足关键业务功能恢复运作要求的场地 (4) 以上场地应保持 7×24 运作

续表

	要　素	要　求
A. 5. 5	技术支持	在备用场地有： (1) 7×24 专职计算机机房管理人员 (2) 专职数据备份技术支持人员 (3) 专职硬件、网络技术支持人员
A. 5. 6	运行维护支持	(1) 按介质特征对备份数据进行定期的有效性验证 (2) 有介质存取、验证和转储管理制度 (3) 有备用计算机机房运行管理制度 (4) 有硬件和网络运行管理制度 (5) 有实时数据备份系统运行管理制度
A. 5. 7	灾难恢复预案	有相应的经过完整性测试和演练的灾难恢复预案

(6) 第六级：数据零丢失和远程集群支持。

第六级相对于第五级的实时数据复制而言，要求实现远程数据实时备份，实现零丢失；备用数据处理系统具备与生产数据处理系统一致的处理能力并完全兼容，应用软件是集群的，可以实现无缝切换，并具备远程集群系统的实时监控和自动切换能力；对于备用网络系统的要求也加强，要求最终用户可通过网络同时接入主、备中心；备用场地还有 7×24 小时专职操作系统、数据库和应用软件的技术支持人员，具备完善、严格的运行管理制度。具体技术和管理支持如表 3.7 所示。

表 3.7　灾难恢复第六级要求

	要　素	要　求
A. 6. 1	数据备份系统	(1) 完全数据备份至少每天一次 (2) 备份介质场外存放 (3) 远程实时备份，实现数据零丢失
A. 6. 2	备用数据处理系统	(1) 备用数据处理系统具备与生产数据处理系统一致的处理能力并完全兼容 (2) 应用软件是集群的，可以实现无缝切换 (3) 具备远程集群系统的实时监控和自动切换能力
A. 6. 3	备用网络系统	(1) 配备与生产系统相同等级的通信线路和网络设备 (2) 备用网络处于运行状态 (3) 最终用户可通过网络同时接入主、备中心
A. 6. 4	备用基础设施	(1) 有符合介质存放条件的备用场地 (2) 有符合备用数据处理系统和备用网络设备运行要求的场地 (3) 有满足关键业务功能恢复运作要求的场地 (4) 以上场地应保持 7×24 运作
A. 6. 5	技术支持	在备用场地有： (1) 7×24 专职计算机机房管理人员 (2) 7×24 专职数据备份技术支持人员 (3) 7×24 专职硬件、网络技术支持人员 (4) 7×24 专职操作系统、数据库和应用软件技术支持人员

续表

	要　　素	要　　求
A.6.6	运行维护支持	(1) 按介质特征对备份数据进行定期的有效性验证 (2) 有介质存取、验证和转储管理制度 (3) 有备用计算机机房运行管理制度 (4) 有硬件和网络运行管理制度 (5) 有实时数据备份系统运行管理制度 (6) 有操作系统、数据库和应用软件运行管理制度
A.6.7	灾难恢复预案	有相应的经过完整性测试和演练的灾难恢复预案

通过分析以上灾难恢复的级别，一个完整的容灾系统应该具有以下几个组成部分：

(1) 本地的高可用系统：确保本地发生局部故障或单点故障时的系统安全；

(2) 数据备份系统：用于抗御用户误操作、病毒入侵、黑客攻击等威胁；

(3) 数据远程复制系统：保证本地数据中心和远程备援中心的数据一致；

(4) 远程的高可用管理系统：实现远程广域范围的数据管理，它基于本地的高可用系统之上，在远程实现故障的诊断、分类并及时采取相应的故障管理措施。

4. 容灾系统的系统结构

容灾系统的系统框架如图 3.7 所示。

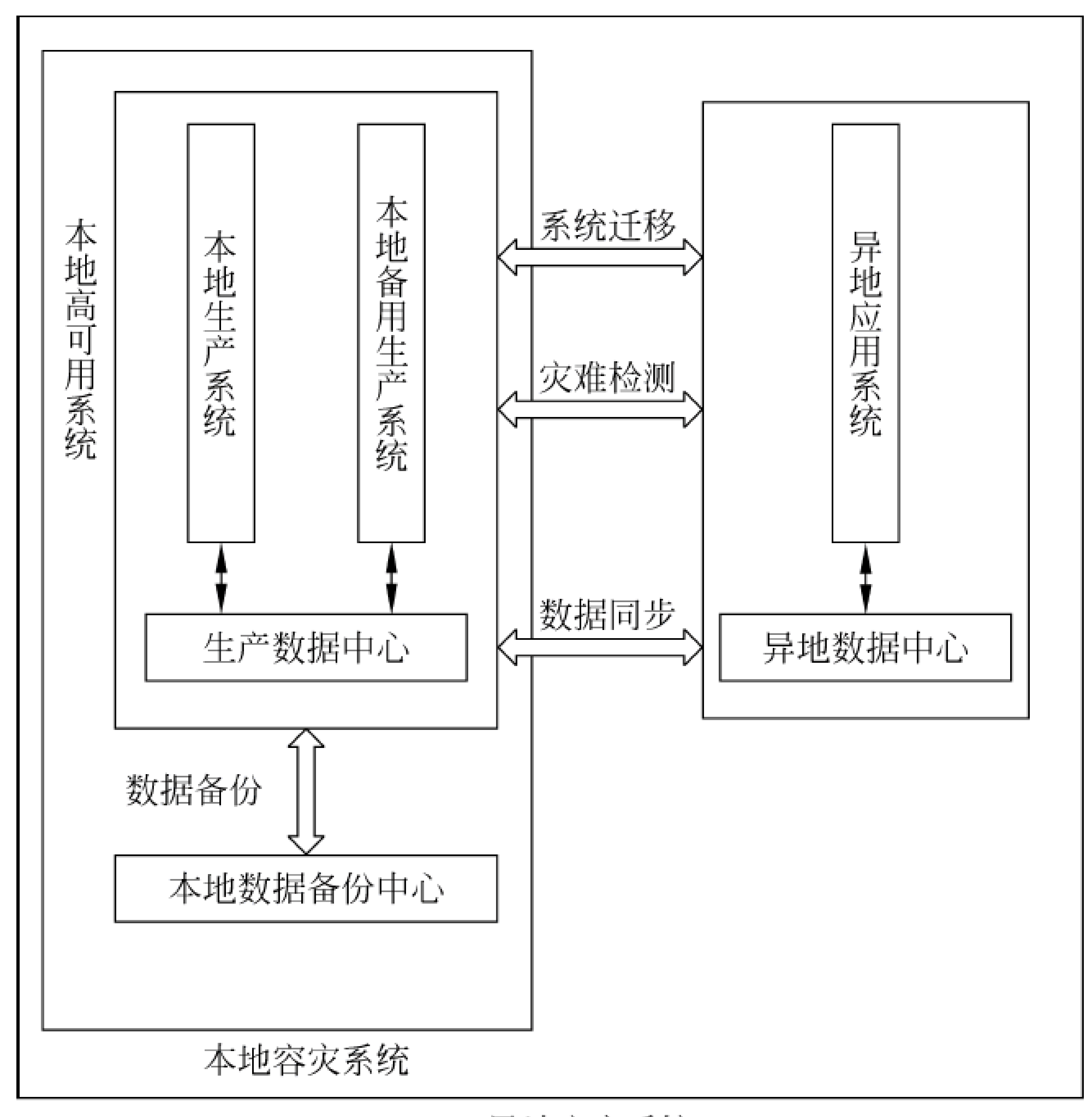


图 3.7　容灾系统的系统框架

容灾系统的主要作用是保证数据完整性和业务连续性，数据完整性是业务连续性的基础。一个完整的容灾系统，应该由本地生产系统、本地备用生产系统、生产数据中心、本地备份数据中心、异地应用系统、异地数据中心 6 部分组成，本地生产系统、本地备用生产系统和

生产数据中心组成了高可用系统，根据需求，可以使用其中的某几部分组成不同级别的容灾系统。

使用本地高可靠系统和本地数据备份中心可建立本地容灾中心，能够容忍硬件毁坏等灾难造成的单点失效，而对于火灾、大楼倒塌等大规模灾难却无能为力。使用本地高可用系统、本地备用数据中心和异地数据中心可以建立异地数据容灾系统。使用本地高可靠系统、本地备用数据中心、异地应用系统和异地数据中心可以建立异地应用容灾系统，而根据异地备份中心与本地系统距离的远近，系统所能容忍的灾难也不相同，如果异地数据备份中心与本地系统在100km之内，可以容忍火灾、停电、建筑物倒塌等灾难；如果达到了几百千米，可以容忍地震、水灾等大规模、大范围的灾难。

本地系统与异地系统的数据同步方式也有很多种选择，例如对于异地数据容灾系统，数据同步方式可以选择用运输工具运输到异地数据中心，也可以选择同步或者异步的方式直接由生产数据中心复制到异地数据中心；而对于异地应用容灾系统，就只能选择同步或者异步的方式直接由生产数据中心直接复制到异地数据中心。在这个容灾系统结构中，本地数据中心需要及时地将数据复制到异地数据中心，并要保证数据的完整性和可用性。而为使异地系统能够及时地发现本地系统的灾难，就需要进行灾难检测，保证异地系统能够及时发现灾难，并能及时地替换本地系统，也就是将本地系统的业务迁移到异地系统，从而保证业务的连续性。

5. 灾难恢复的指标

在灾难恢复领域，除了等级划分，还提供了用于量化描述灾难恢复目标的最常用的恢复目标指标：RPO(恢复点目标)和RTO(恢复时间目标)。

(1) 恢复点目标(Recovery Point Objective，RPO)：灾难发生后，系统和数据必须恢复到的时间点要求。它代表了灾难发生时允许丢失多长时间的数据量。例如，1小时的RPO指灾难发生后容灾系统能够对灾难发生1小时前的所有数据进行恢复，但这1小时的数据可能会丢失。

(2) 恢复时间目标(Recovery Time Objective，RTO)：灾难发生后，信息系统或业务功能从停顿到必须恢复的时间要求，它代表了系统恢复的时间。

PRO描述的是数据丢失指标，而RTO描述的是服务丢失指标，二者没有必然的关联性。实际中可根据RPO和RTO的要求规划建设容灾备份系统。

3.7.3　容灾系统关键技术

容灾系统所包含的关键技术包括数据存储技术、远程镜像技术、灾难检测、系统迁移等。

1. 数据存储技术

容灾系统需要存储的数据量庞大，为了提高备份的效率，出现了很多新的备份技术，在很大程度上提高了备份速度，目前采用的备份技术主要有以下几种：

1) 直接附加存储(Direct Access Storage，DAS)

传统的直接附加存储结构中，将存储设备(如磁盘、阵列)通过SCSI接口附加在服务器上，由服务器提供存储设备的管理和对外服务。这种存储结构价格比较便宜，但是支持的存储容量有限制，每条并行的SCSI总线最多只能支持15个磁盘阵列，当业务量非常大、需要

存储的数据非常多时,这种存储结构就不大适用了。并且客户每次访问存储设备中的数据时,数据需要在存储设备和服务器之间多次转发,尽管服务器并不关心数据内容,通常也不对数据本身进行处理,但数据请求和传送都需要服务器的介入,存储容量扩大后,对同一台服务器进行访问,容易形成访问瓶颈。

2) 网络附加存储(Network Attached Storage,NAS)

网络附加存储是一种以数据为中心的存储结构,存储子系统不再附属于某个服务器,而是通过专门系统的定制,将通用服务器上的无关功能去掉,只保留存储相关功能,可以看成是一台专门负责存储的“瘦”服务器,具有比 DAS 更高的读写性能。NAS 将存储设备通过网络协议控制器直接连接在局域网上,通过 NAS 内部的文件管理系统对外提供服务。

将 NAS 设备连接到网络上非常方便。NAS 设备提供 RJ-45 这样的网络物理接口和单独的 IP 地址,可以将其直接挂接在主干网的交换机或其他局域网的 Hub 上,通过简单的设置(如设置机器的 IP 地址等)就可以在网络即插即用地使用,而且进行网络数据在线扩容时也无须停顿,从而保证数据流畅存储。与传统的服务器或 DAS 存储设备相比,NAS 可以拥有更大的存储空间和相对低廉的价格。

由于普通的 LAN 不是针对存储应用设计的专用网络,而且目前大部分 LAN 还是使用 10Mb/s 或 100Mb/s 的传输速率连接,加上 LAN 上又有大量计算机,因此网络有限的带宽要面对大量的传输需求,NAS 存储设备所能分到的带宽必然有限。这就造成了 NAS 的最大缺点,传输速率慢且不稳定,这在进行备份或者大文件存取时将花费大量的时间。

3) 存储区域网络(Storage Area Network,SAN)

SAN 的设计思想实际上很简单,就是建立一个单独的网络系统,采用适合数据传输和管理特点的物理、链路、网络传输等各层协议,专门用于存储的管理和数据交换。目前该网络使用 FC(Fiber Channel)协议。该网络只用于存储,不会有其他服务的数据流在上面传输,可以做到独享带宽;而且因为光纤通道本身具有的高传输速率,使得 SAN 网络的传输速率可以达到 200Mb/s 或者更高,同时还可以保证数据传输速率的稳定性。但也正是由于受到 SAN 使用专用网络拓扑结构和不同于一般网络传输协议的限制,SAN 的设备仅能做到与连接在 SAN 上的服务器间的直接访问,而在 LAN 上的客户端是无法直接访问 SAN 的设备的,必须通过服务器间接访问。由于 SAN 的硬件设备价格昂贵,而且,SAN 作为一种专用的存储网络,需要培训专门的人员来管理,这使得 SAN 的总体拥有成本居高不下,使得很多希望使用 SAN 的企业望而却步,转而使用性能较差的 NAS,因而 SAN 的普及和使用受到较大影响。

4) 基于 IP 的存储网络和 iSCSI

为了解决前面提到的 SAN 应用带来的问题,又出现了基于 IP 的存储网络,IP 存储网络可以说是结合了 NAS 和 SAN 两者的优点:一方面它采用 TCP/IP 作为网络协议,使得它具有 NAS 易于访问的特点;另一方面它又有独立专用的存储网络结构。因此,基于 IP 的存储网络可以使用目前应用广泛的以太网(Ethernet)技术和设备来构建专用的存储网络,通过使用 Ethernet 的设备,其成本与 FC SAN 相比大为降低,而且还保持有 SAN 的传输速率高且稳定的优点。以上两点可以说是基于 IP 的存储网络技术的两个最大的优势。

IP-SAN 最大的问题是它的性能能否达到 FC-SAN 的标准。Ethernet 虽然已经出现了很长时间,但由于 Ethernet 已拥有的大量用户和巨大市场,各个厂家不会放弃它,Ethernet

仍然具有很大的发展潜力。虽然 FC 协议也在持续不断的发展，但是 FC-SAN 的用户数量和市场范围都远无法和 Ethernet 相比，其发展动力也就不如 Ethernet 大。Ethernet 速度目前已经有了数量级的提高，千兆级 Ethernet 也已经投入使用，但目前主要用于服务器端或构建主干网。千兆级 Ethernet 在速度上已经可以和 FC 相比了，其传输介质可以使用光纤、无屏蔽双绞线等多种传输介质，其价格却不像 FC 那么昂贵。下一步 Ethernet 的速度会达到 10Gb/s，而 FC 的下一个目标只是制造和推广 2Gb/s 的产品。

目前基于 IP 的存储网络的核心技术是 iSCSI，这是一种开放协议，其基本架构是在 SCSI 的数据包上加上 TCP/IP 协议，由于加入了 TCP/IP 协议，iSCSI 协议可以使 SCSI 数据包在普通的 IP 网络上传输。iSCSI 协议与 FC 协议没有任何联系，该协议的最终目的是取代 FC 协议在 SAN 中的位置。

2. 远程镜像技术

数据备份技术通常是在本地节点进行的备份操作，备份间隔的单位通常为天或月，生成静态的文件，可以经过压缩等处理静态保存，在灾难发生时能够从备份中将数据恢复出来。例如保存于光盘、磁带、硬盘等数据备份介质上的数据，需要经过恢复技术配合合适的系统硬件环境才能恢复出来供业务系统使用。

而在容灾系统中的远程镜像则是将数据实时或准实时复制到异地节点，这是一个动态的过程，数据是在不断更新的，复制的数据在异地节点上保持原来的数据形态，与本地节点的数据保持基本一致性，可以不经过恢复技术就可直接使用。

镜像是在两个或多个磁盘或存储系统上产生同一个数据镜像视图的一个信息存储过程，一个叫主镜像系统，另外一个叫从镜像系统。按主、从镜像存储系统所处的位置可分为本地镜像和远程镜像，远程镜像是容灾备份的核心技术。按请求镜像的主机是否需要远程镜像站点的确认信息，又可分为同步远程镜像和异步远程镜像。

同步远程镜像中每一步本地 I/O 事务均需等待远程复制的完成方予释放，这种方式的远端数据与本地数据完全同步，但由于数据复制过程中存在时延，本地 I/O 访问效率下降，所以只限于在相对较近的距离上应用（一般专线连接在 60km 以内，常见于同城系统），同时，这种复制技术还受到带宽因素的制约，若远程的 I/O 带宽较窄时，会显著拖慢主数据中心的 I/O，影响系统性能。但同步镜像使远程拷贝总能与本地机要求复制的内容相匹配，当主站点出现故障时，用户和应用程序换到一个代替站点后，远程的副本可以继续执行操作。

异步远程镜像保证在更新远程存储视图前完成向本地存储系统的基本输入输出操作，而由本地存储系统提供给请求镜像服务器的操作完成确认信息，不需要等待远程存储系统提供操作完成确认信息，这使得本地系统性能受到很小的影响。但是，许多远程的从属存储子系统的写操作没有得到确认，当某种因素造成数据传输失败时，可能出现数据一致性问题。为了解决数据一致性问题，目前大多采用延迟复制的技术，它可以在确保本地数据完好无损后进行远程数据更新。

3. 快照

远程镜像技术往往同快照技术结合起来实现数据信息的远程备份，即通过镜像把数据备份在远程存储系统中，再借助快照技术把远程存储系统中的信息备份到远程的磁带库、光盘库中。快照是通过软件对要备份的磁盘子系统的数据快速扫描，在正常应用进行的同时

实现对数据的完全备份。它可使用户在正常应用不受影响的情况下实时提取当前在线数据,其备份窗口接近于零,可大大增加系统应用的连续性,为实现系统真正的不间断运转提供了保证。

4. 灾难检测

对于火灾、地震等大规模灾难,当然可以依靠人为确定,但是对于停电、硬件毁坏等很难觉察到的灾难就不能仅仅依靠人去发现。现在对灾难的发现方法一般是通过心跳技术和检查点技术,这种技术在高可靠性集群中应用很广泛。对于异地容灾,备份生产中心和主生产中心可能相隔千里,这时候因为网络延迟较大或者其他原因,可能会影响心跳检测的效果,因此如何对现有的检测技术进行改进,以适应广域网的要求,将是实现高效的远程容灾系统的基础。

心跳技术,就是每隔一段时间都要向外广播自身的状态(通常为“存活”状态),在进行心跳检测时,心跳检测的时间和时间间隔是关键问题,如果心跳检测得太频繁,将会影响系统的正常运行,占用系统资源;如果间隔时间太长,则检测就比较迟钝,影响检测的及时性。检查点技术又称为主动检测,就是每隔一段时间周期,就会对被检测对象进行一次检测,如果在给定的时间内,被检测对象没有响应,则认为检测对象失效。与心跳技术相同,检测点技术也受到检测周期的影响,如果检测周期太短,虽然能够及时发现故障,但是给系统造成很大的开销;如果检测周期太长,则无法及时发现故障。

为了能够实现异地容灾系统,就必须建立广域网上的分布式可靠性系统,这就需要有高效的故障检测系统,能够及时地发现故障,及时切换。而对于广域网来说属于异步的系统,没有同步的时钟、没有可靠的传输通道,如何在异步的分布式模型中实现可靠高效的故障检测将是建立异地容灾系统的基础。

5. 系统迁移

在发生灾难时,为了能够保证业务的连续性,必须能够实现系统透明迁移,也就是能够利用备用系统透明地代替生产系统。对于实时性要求不高的容灾系统,通过 DNS 或者 IP 地址的改变来实现系统迁移便可以了,但是对于可靠性、实时性要求较高的系统,就需要使用进程迁移算法,进程迁移算法的好坏对于系统迁移的速度有很大影响,现在该算法在分布式系统和集群中得到了广泛的运用,并发挥着重大作用,也有很多研究对该算法的性能进行了改进。

进程迁移算法在目前主要有贪婪拷贝算法、惰性拷贝算法和预拷贝算法。贪婪拷贝算法简单、易于实现,但是延时较长,并且冗余数据会造成较大的网络延迟;惰性拷贝的延迟小、网络负担小,但是对原主机具有依赖性、可靠性差;预拷贝算法将信息分两次拷贝,使得传输时间反而增长。现在的进程迁移算法都是应用于本地集群的,如何在远距离容灾系统中实现高效的进程迁移,就必须对进程迁移算法进行改进,使它能够适应广域网复杂的环境。

3.8 小　　结

物理安全措施用以保护计算机网络设备、设施以及其他介质免遭地震、水灾、火灾、电磁污染等环境事故以及人为操作失误或错误及各种计算机犯罪行为导致的破坏,是信息系统

安全的基础。广义的物理安全主要涉及设备安全、环境安全、媒体安全和系统安全，设备安全主要是指设备的防盗、防毁、防电磁信息泄露、防线路截获、抗电磁干扰；环境安全的主要目的是保护信息系统免受水、电、有害气体、地震、雷击和静电的危害；介质安全的主要目的是保护存储在介质上的信息，包括介质的防盗、介质的防毁，如防霉和防砸等；系统安全是指为保证信息系统的安全可靠运行，降低或阻止人为或自然因素从物理层面对信息系统保密性、完整性、可用性带来的安全威胁，从系统的角度采取的适当安全措施。可靠性是衡量信息安全的主要指标，为了提高系统可靠性可采用避错、容错、容灾备份技术。

习　题

一、填空题

1. ________ 简称为 TEMPEST（Transient ElectroMagnetic Pulse Emanations Standard Technology）技术。

2. 提高系统可靠性的方法有________、________和________。

3. 物理安全包括________、________、________和________。

4. 环境安全面临的安全威胁有________、________和________等。

5. 容错技术包括________、________、________和________。

6. 按照建立容灾系统目标的不同，容灾备份系统可以分为两种，分别是________和________。

7. 用于量化描述灾难恢复目标的最常用的恢复目标指标是________和________。

8. 硬件冗余有三种基本形式，分别是________、________和________。

9. 软件容错技术的方法主要有________和________。

二、简答题

1. 环境可能对计算机安全造成哪些威胁？如何防护？

2. 计算机哪些部件容易产生辐射？如何防护？

3. TEMPEST 技术的主要研究内容是什么？

4. 计算机设备防泄漏的主要措施有哪些？它们各自的主要内容是什么？

5. 为了保证计算机安全稳定地运行，对计算机机房有哪些主要要求？机房的安全等级有哪些？根据什么因素划分？

6. 灾难恢复的指标是什么，分别代表什么含义？

7. 国际上如何划分灾难恢复的等级？

8. 按照建立容灾系统目标的不同，容灾备份系统可以分为几种，分别适用于什么场合？

第4章　身份认证

身份认证是信息系统的第一道安全防线,其目的是确定用户的合法性,阻止非法用户访问系统。身份认证对确保信息系统和数据的安全是极其重要的,可以通过验证用户知道什么、用户拥有什么、用户的生理特征等方法来进行用户的身份认证。

4.1节给出了身份认证的定义和主要的认证方式,4.2节介绍了常用的口令认证机制并分析了该机制存在的安全隐患和相应的增强机制,4.3节介绍了一次性口令认证的原理和实现机制,4.4节介绍了智能卡认证方式,详细介绍了USB Key认证原理,4.5节简要介绍了基于生物特征的认证方式,4.6节介绍了三种身份认证协议,4.7节简要介绍了零知识证明。

4.1　概　　述

身份认证是验证主体的真实身份与其所声称的身份是否相符的过程,它是信息系统的第一道安全防线,如果身份认证系统被攻破,那么信息系统所有其他安全措施将形同虚设,因此,身份认证是信息系统其他安全机制的基础。

身份认证包括标识与鉴别两个过程。标识(Identification)是系统为区分用户身份而建立的用户标识符,一般是在用户注册到系统时建立,用户标识符必须是唯一的且不能伪造。将用户标识符与用户物理身份联系的过程称为鉴别(Authentication),鉴别要求用户出示能够证明其身份的特殊信息,并且这个信息是秘密的或独一无二的,任何其他用户都不能拥有它。

【例4-1】 用户Alice在某信息系统中的标识符为abtoklas,跟用户标识符相关的认证信息是用户口令,该口令存储在信息系统中,只有Alice和系统知道,如果没有人能获取或猜测Alice的口令,那么标识符和密码的组合可以认证用户的身份。

常见的身份认证方式有以下几种:

(1) 利用用户所知道的东西(Something the User Knows),如口令、PIN码或者回答预先设置的问题。

(2) 利用用户所拥有的东西(Something the User Possesses),如智能卡等物理识别设备。

(3) 利用用户所具有的生物特征(Something the User is or How He/She Behaves),如指纹、声音、视网膜扫描、DNA等。

这几类身份认证方式各有利弊。第一类方法最简单,系统开销小,但是最不安全;第二类安全性比第一类高,但是认证系统相对复杂;第三类的安全性最高,但是涉及更复杂的算法和实现技术。前两类身份认证技术起步较早,目前相对成熟,应用也比较广泛,第三类技术由于它在安全性上的优势正在迅速发展。

一般情况下,可以通过多个因素(Multi-Factor)共同鉴别用户身份的真伪,例如我们在银行ATM机上取款需要插入银行卡,同时需要输入银行卡密码,这就采用了双因素认证,

鉴别的因子越多,鉴别真伪的可靠性就越大。当然,在设计鉴别机制时需要考虑认证的方便性和性能等综合因素。

下面逐一讨论以上三种形式的身份认证技术。

4.2　基于口令的认证

4.2.1　口令认证过程

口令是通信双方预先约定的秘密数据,基于口令的认证方式是最常用的一种身份认证技术,这种认证方法简单易行,不需投入太多就可以实现,广泛应用于操作系统、网络、数据库和应用程序中。

在口令认证中,用户事先在服务器上进行注册,建立自己的用户账号,包括用户标识符和口令,这些信息存储在服务器口令表中,只有用户自己和服务器知道,通常情况下,只要用户保持口令的保密性,非授权用户就无法使用该用户的账户。图 4.1 描述了口令认证的过程,用户在登录界面输入用户标识符和口令,服务器在接收到客户端认证请求后,跟数据库口令表中存储的信息进行比对,如果找到匹配项则认证通过,否则返回认证失败信息。

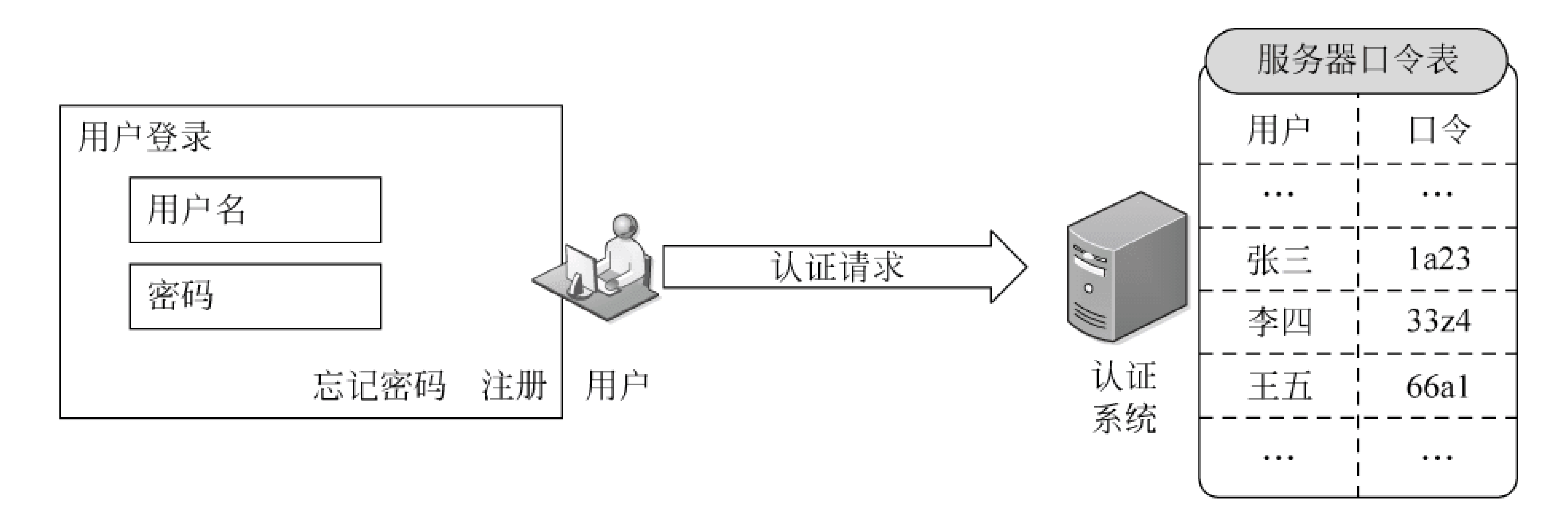


图 4.1　基于口令的身份认证过程

图 4.1 所示的身份认证方式的优点是简单易用,在安全性要求不高的情况下易于实现,但是该机制存在严重的安全问题,主要包括以下几个。

1. 口令质量不高

通常用户为了记忆、使用方便,会采用与自己周围事物相关的单词或数字作为口令,这些类型的口令容易破解,常被称为弱口令,常见的弱口令类型有:

(1) 用户名或用户名的变形;

(2) 电话号码、执照号码等;

(3) 一些常见的单词;

(4) 生日;

(5) 长度小于 5 的口令;

(6) 空口令或默认口令;

(7) 上述词后加上数字。

针对弱口令的攻击主要有字典攻击和穷举攻击(暴力破解)。

(1) 字典攻击

由于大部分人选用的密码都是与自己周围事物有关的单词或数字,攻击者利用各种手段收集用户可能使用的口令构成口令字典,借助于口令破解工具逐一尝试字典中的口令,实现口令的破解。

【例 4-2】 密码字典是一个字符串表,表中的每一项都是一个可以作为密码的字符串,但不需要对该字符串做出解释,所以密码字典更像一个词汇列表,图 4.2 就是典型的密码字典的"庐山真面目"。

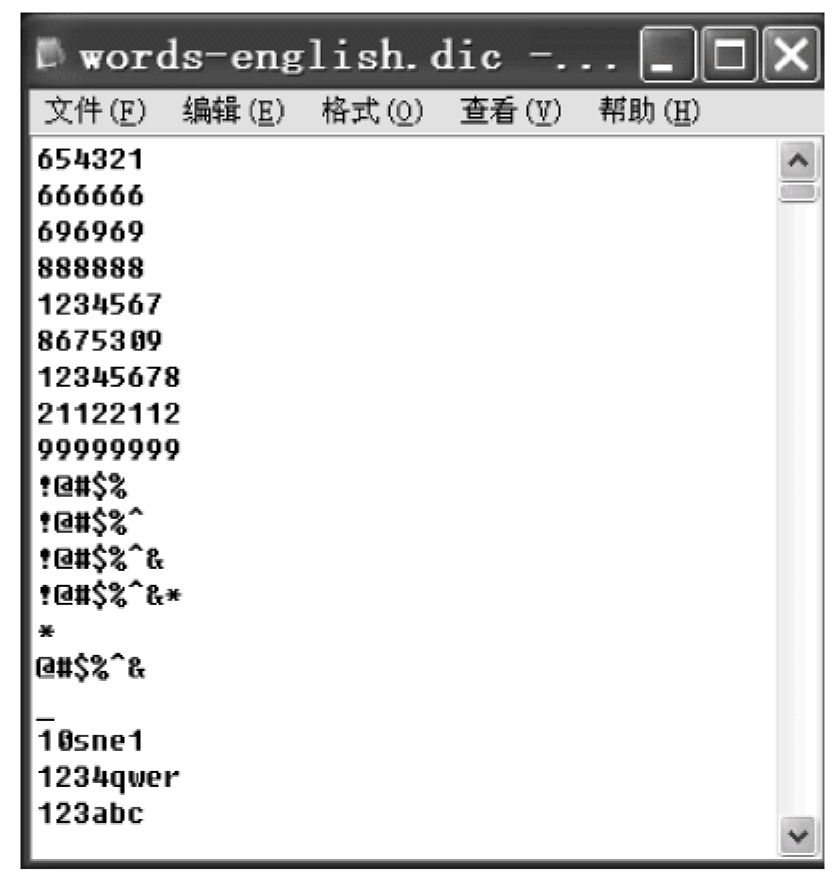


图 4.2　密码字典

(2) 穷举攻击(暴力破解)

如果把字符串的全集作为字典进行攻击称为穷举攻击,通俗地说就是尝试所有可能的口令,直到破解成功,这是一种最原始、最粗犷的方式,也称为暴力破解。当用户的密码比较短,很容易被穷举。

【例 4-3】 在 Windows(Windows NT 和 Windows 2000/XP) 平台上,口令文件是% systemroot% \system32\config 中一个名为"SAM"的文件;在 UNIX/Linux 平台上,口令文件是/etc/passwd 或者/etc/shadow。一旦攻击者获得了以上信息,就会尝试各种口令攻击工具来进行破解。常用的两个口令破解工具如"John the Ripper"和"Lophtcrack",分别用于破解 UNIX/Linux 系统和 Windows NT/2000/XP 下的口令文件,这两个口令破解工具的原理就是字典攻击和穷举攻击。

2. 口令明文存储

认证服务器端如果采用明文方式存储口令,一旦攻击者成功访问数据库,则可以得到所有用户的用户名、口令信息。

【例 4-4】 2011 年 12 月 21 日,中国最大的软件开发者技术社区 CSDN 后台数据库被盗,由于明文存储,642 万多个用户的账号、口令等信息被泄露。

3. 口令嗅探和重放攻击

一些信息系统对传输的口令没有进行加密处理,攻击者可以轻易得到口令的明文。但是即使口令经过加密也难以抵抗重放攻击,因为攻击者可以将截获的加密信息直接发送给认证服务器,因为这些加密信息是合法有效的,服务器同样也能认证通过。

4. 攻击者运用社会工程学、键盘记录器等获取口令

4.2.2　口令认证安全增强机制

为此,需要采取措施对口令认证机制进行安全增强。

1. 提高口令质量

口令认证的安全性很大程度上取决于口令质量,它涉及以下几点。

(1) 增加口令的复杂度和长度。

增加口令复杂度需要增大口令的字符空间,口令的字符空间不要仅限于常见的 26 个小写字母、0~9 十个数字,要扩大到所有可被系统接受的字符,如 26 个大写字母,特殊字符

@、#、%、* 等，选用的口令最好是字母、数字、特殊字符的组合。同时还需注意口令的长度，选择长口令可以增加破解的时间，假定口令长度为 4，口令字符空间为 95，则组成的所有可能口令的数目为 95^4，对于一台处理性能为每秒 100 万条指令的高性能计算机来说，采用穷举攻击破译口令大约需要 2min，而当口令长度增长到 6，采用同样的计算机，破译时间增长到 8 天，如表 4.1 所示，由此可见，口令破解的难度随口令长度呈指数级增长。满足复杂度和长度要求的口令为强口令，为了增强口令认证的安全性，需要采用不容易被破解的强口令。

表 4.1　可通过键盘输入的字符分类

分　类	个　数	解　释
小写字母	26	abcdefghijklmnopqrstuvwxyz
大写字母	26	ABCDEFGHIJKLMNOPQRSTUVWXYZ
数字	10	0123456789
其他	33	! @ # $ %^& * ()-=_+[]{}|;':",.<>/?'~
总计	95	

（2）在用户使用口令登录时，还可以采取更加严格的控制措施：

① 登录时间限制。例如，用户只能在某段时间内才能登录到系统中。

② 限制登录次数。例如，如果有人连续多次登录失败，则认证系统拒绝再次认证请求，这样可以防止字典攻击和暴力破解。

③ 尽量减少会话透露的信息。例如，登录失败时不提示是用户名错还是口令错，使外漏的信息最少。

2. 口令加密存储

采用明文方式存储口令有很大风险，任何人只要得到存储口令的数据库，就可以得到全体合法用户的口令。

目前常用的方法是服务器口令文件中存储用户名（标识符）和口令的散列值，当用户登录时，用户输入口令 x，系统计算 Hash(x)，然后与口令文件中相应的散列值进行比对，匹配则允许登录，否则拒绝登录。

由于在口令文件中存储的是口令的散列值，黑客即使得到口令文件，通过散列值想要计算出原始口令在计算上也是不可能的，这就相对增加了安全性。

3. 口令加密传输

在通信链路上，如果口令以明文传输，黑客可以采用网络监听工具对通信内容进行网络嗅探，盗取传输的用户名和口令信息，为了应对这种网络嗅探行为，可以对传输的口令信息进行加密，例如利用散列函数对传输的口令进行加密处理。

但是即使对传输的口令进行加密，攻击者仍可以冒充用户身份。利用截获的密文信息，攻击者可以构造新的登录请求并将其提交到同一服务器，服务器不能区分这个登录请求是来自合法用户还是来自攻击者，这种攻击称为重放攻击。传统的口令认证之所以无法抵御重放攻击，主要原因是通常使用的计算机口令是静态的，也就是说在一定时间内是不变的，而且可重复使用，这就让攻击者有机可乘，利用截获的信息可以重复登录，冒充用户身份执行非法操作。为了解决这个问题，需要采用一次性口令技术（又称为动态口令），同一口令不能重复认证，即使攻击者截获了传输的认证数据包，采用重放攻击再次向服务器发送，由于

数据包中的口令已经被使用过，无法验证通过，采用动态口令可以有效地抵抗重放攻击。

4.3 一次性口令的认证

20世纪80年代初，针对静态口令认证的缺陷，美国科学家Leslie Lamport首次提出了利用散列函数产生一次性口令(One Time Passwd,OPT)的思想，即在用户的每次登录过程中，加入不确定因素以生成动态变化的示证信息，从而提高登录过程的安全性。由贝尔通信研究中心于1991年开发的S/KEY是一次性口令的首次实现。

目前一次性口令技术已经得到了广泛应用，比较简易的实现有短信密码、验证码、口令卡等。在短信密码中，身份认证系统以短信形式发送随机的6/8位密码到客户的手机上，客户在登录或交易认证时输入此动态密码，从而确保系统身份认证的安全。验证码通常称为全自动区分计算机和人类的图灵测试(Completely Automated Public Turing test to tell Computer and Humans Apart,CAPTCHA)，是一种主要区分用户是计算机和人的自动程序。这类验证码的随机性不仅可以防止口令猜测攻击，还可以有效防止攻击者对某一个特定注册用户用特定程序进行不断的登录尝试，例如刷票、恶意注册、论坛灌水等。

【例4-5】 使用口令卡实现一次性口令认证。

如图4.3所示，是应用于网上银行的口令卡，卡的一面以矩阵的形式印有若干字符串，初始时有覆膜。不同的账号对应的口令卡不同，用户在网上银行进行对外转账或缴费等支付交易时，网上银行系统随机给出一组口令卡坐标，客户根据坐标从卡片中找到对应的口令组合并输入到网上银行系统中，网上银行系统据此来对用户进行身份鉴别。

基于口令卡的鉴别过程如下：

(1) 认证端系统的数据库中存放用户的账号和对应的口令卡内容。

(2) 用户向认证端发送认证请求，该请求中包含用户标识符信息。

(3) 认证端检查用户标志符是否有效，如果无效，则向用户返回错误信息，结束鉴别过程，如果有效，则进入下一步。

(4) 认证端生成两个随机坐标(也称挑战)，并通过网络传输给用户端。

(5) 用户根据坐标利用口令卡查出对应的6位口令，并将查找的结果发送给认证端。

(6) 认证端利用相同的口令卡验证用户发送过来的口令，如果一致则鉴别通过，否则返回鉴别失败信息。

	B	C	G	J	K	P	S	U	V	X
1	883	814	885	521	362	234	816	646	742	028
2	306	521	259	029	856	138	342	657	568	738
3	291	051	611	850	797	555	772	692	447	536
4	206	813	949	309	894	785	560	289	547	437
5	041	343	244	798	499	388	964	880	823	521
6	318	119	661	878	503	517	955	281	616	567
7	180	493	930	965	638	056	609	356	611	920
8	592	133	694	827	745	196	434	339	940	130

图4.3 口令卡

采用口令卡的认证方式，用户每次输入不同的动态口令，防止了重放攻击。用户只要保管好手中的口令卡，就能较好地确保资金交易的安全，但是口令卡所提供的一次性口令是有限的，一旦口令卡中的口令使用完后，需要重新换卡。该方法对用户端要求比较低，不要求用户端进行数据加密，攻击者经过网络嗅探，可重构口令卡坐标数据，这种认证方式安全系数较低，因此，一般口令卡对网上交易有金额限制，如果要进行大额交易，建议使用安全性更强的 U 盾之类的口令令牌。

上述一次性口令产生机制比较简单，动态口令数目有限，并且传输过程中没有进行加密处理，更为安全的一次性口令是以密码学为基础产生的。根据动态因素的不同，主要分为两种实现技术：同步认证技术和异步认证技术。其中同步认证技术又分为基于时间同步认证技术(Time Synchronous)和基于事件同步认证技术(Event Synchronous)；异步认证技术即为挑战/应答认证技术(Challenge/Response)。

1. 基于时间同步的认证技术

在一次性口令生成机制中，时间同步方案原理较为简单，该方案产生一次性口令的动态因素是登录时间，为了使服务端能产生跟用户相同的口令以验证用户的身份，该方案要求用户和认证服务器的时钟必须严格一致，用户持有称为动态令牌(如图 4.4 所示)的专用硬件，令牌内置电源、同步时钟、秘密密钥和机密算法。动态令牌根据同步时钟和密钥每隔一个时间单位(如 1min)产生一个动态口令，用户登录时输入动态令牌显示的当前密码发送给认证服务器，认证服务器根据当前时间和密钥副本采用相同的算法计算出口令，最后将认证服务器计算出的口令和用户发送的口令相比较，得出授权用户的结论。



图 4.4　动态口令牌

除了采用动态令牌硬件产生一次性口令之外，随着手机的普及，出现了手机令牌方式，手机令牌是一种手机客户端软件，基于时间同步方式，每隔 30s 产生一个随机 6 位动态密码，口令生成过程不产生通信及费用，具有使用简单、安全性高、低成本、无须携带额外设备等优势，手机令牌是 3G 时代动态密码身份认证的发展趋势。

时间同步机制具有操作简单、携带方便等优点，使用该方案的难点在于需要解决好网络延迟等不确定因素带来的干扰，使口令在生命周期内顺利到达认证系统。

2. 挑战/应答(Challenge/Response)方式

应用最广泛的一次性口令实现方式是挑战/应答方案，该方案的基本原理为每次认证时，服务器产生一个随机数(又称为挑战)发送给客户端，客户端以该随机数作为不确定因素，用某种单向算法计算出结果作为应答，服务器通过验证应答的有效性来判断客户端身份的合法性。最为经典的挑战/应答方案是 S/Key 协议，S/Key 口令认证方案分为两个过程：注册过程和认证过程。注册过程只执行一次，而认证过程则在用户每次登录时都要执行。

1) S/Key 注册过程

步骤 1：新用户选择将在服务器上注册的用户名 ID、口令 PW，并提交给服务器，服务器为该 ID 选择随机种子 Seed 和最大迭代数 Seq，并将 Seed 和 Seq 发送给用户。

步骤 2：用户收到 Seed 和 Seq 后，对 Seed||PW 进行 Seq 次 Hash 运算 H_{Seq}(Seed||PW)，并将计算结果通过安全的信道发送给认证服务器。

步骤 3：服务器收到 H_{Seq}(Seed||PW)后，将 H_{Seq}(Seed||PW)与对应的用户 ID 保存在认证数据库，同时将最大迭代数减 1 保存。

在完成注册工作后，用户只需记住其登录 ID 和口令 PW 即可。

2）S/Key 认证过程

当用户第 i 次登录时，在用户登录的服务器上当前保存的认证数据有用户 ID、$H_{Seq-i+1}$(Seed||PW)和迭代次数(Seq$-i$)，第 i 次登录认证过程如下，如图 4.5 所示。

步骤 1：用户发送登录请求，并将 ID 发送给服务器，服务器从数据库中取出相对应的种子 Seed 和迭代次数(Seq$-i$)，并将这两个数据传送给用户。

步骤 2：用户收到 Seed 和 Seq$-i$ 后，计算 H_{Seq-i}(Seed||PW)，并将结果发送给认证服务器。

步骤 3：服务器收到认证数据后，计算 H(H_{Seq-i}(Seed||PW))，并与数据库中保存的 $H_{Seq-i+1}$(Seed||PW)相比较。若两者相同，则认证通过，用户成功登录，服务器将收到的 H_{Seq-i}(Seed||PW)替换数据库中的 $H_{Seq-i+1}$(Seed||PW)，以便下一次认证时使用，否则，认证失败，服务器拒绝用户的登录请求。

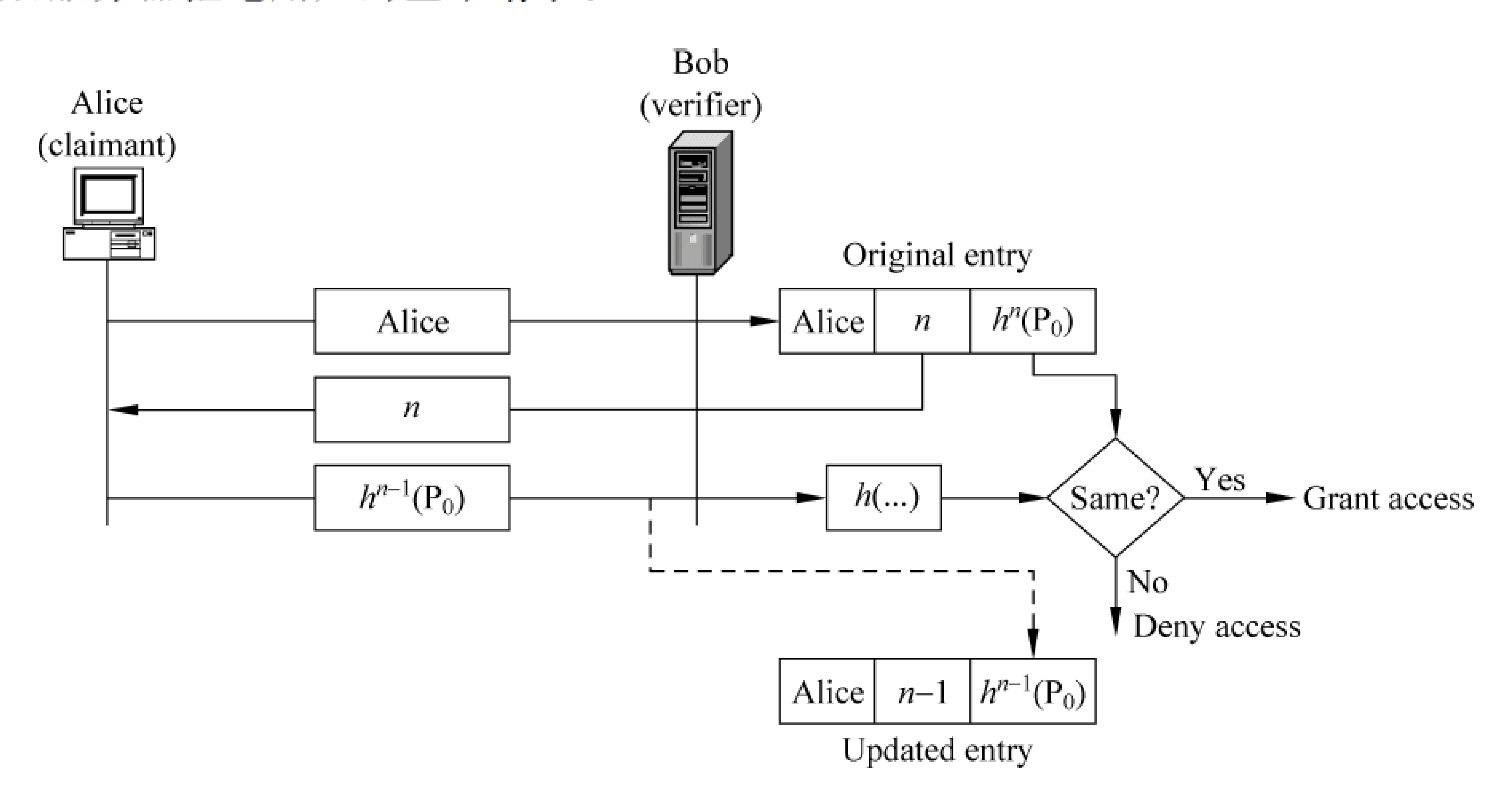


图 4.5 S/Key 认证过程

S/Key 口令序列认证方案在认证过程中只有一次性口令在网络中传输一次，所以该方案基本能满足最初的抵御被动攻击的目标。但是 S/Key 口令认证方案存在“小数攻击”、“协议破坏攻击”、“内部人员攻击”等形式的网络攻击。所谓小数攻击是当用户向服务器请求认证时，黑客截取服务器传来的种子和迭代值，并修改迭代值为较小值，假冒服务器，将得到的种子和较小的迭代值发送给用户。用户利用种子和迭代值计算一次性口令，黑客在此截取用户传来的一次性口令，并利用已知的单向散列函数依次计算较大迭代值的一次性口令，就可以获得该用户后继的一系列口令，进而在一段时间内冒充合法用户而不被察觉，这就是小数攻击。此外，S/Key 口令序列认证方案在执行性能方面还存在运算量大、需要多次散列计算、当迭代次数较小时需要重新进行初始化等不足。

当前挑战应答机制的实现主要是基于对称密码或非对称密码实现的，例如采用对称密码实现挑战应答一次性口令机制的其基本工作工程如下，如图 4.6 所示。

(1) 认证请求。用户端首先向认证端发出认证请求。

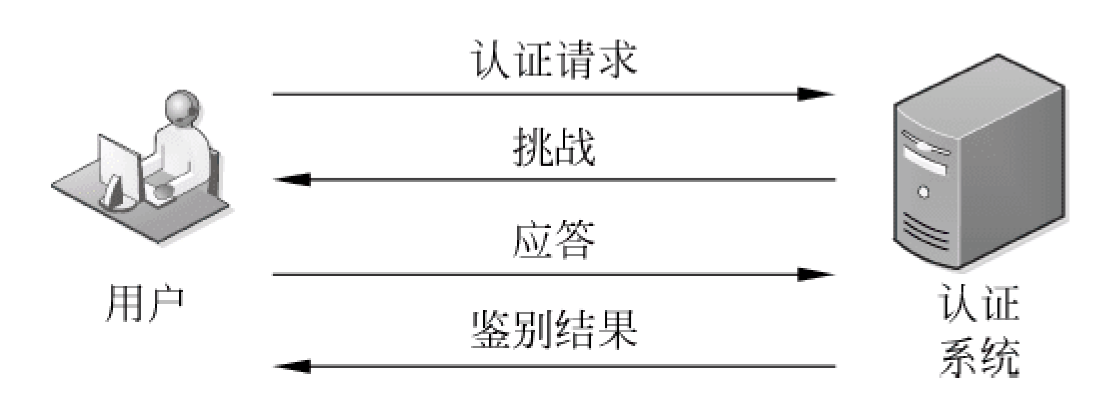


图 4.6　挑战/应答的基本工作过程

(2) 挑战(或质询)。认证端产生一个随机数 X(称为挑战)发送给客户端。同时,认证端根据用户 ID 取出对应的密钥 K,在认证端用加密引擎对发送给客户机的随机数进行运算,得到运算结果 Es。

(3) 应答。客户端程序利用用户和认证端共享的密钥 K 对发送过来的随机数进行加密得到运算结果 Ev,并将此结果作为认证数据发送给认证端。

(4) 鉴别结果。认证端比较 Ev、Es 是否相同,如果相同则该用户为合法用户。

4.4　基于智能卡的认证方式

利用用户所拥有的东西进行身份认证相对复杂、成本高,但这种方法的安全性也比较高,它需要借助于一些物理介质,如磁卡、智能卡等。

磁卡是曾经得到广泛应用的一种用以证实个人身份的手段,由于磁卡仅有数据存储能力,而无数据处理能力,没有对其记录的数据进行保护的机制,因此伪造和复制磁卡比较容易。随着微处理器的发展,出现了智能卡。

智能卡又称 CPU 卡,是一种嵌有单片机芯片的 IC 卡(Integrated Circuit card),在外形上,它将一个集成电路芯片镶嵌于塑料基片中,封装成卡的形式,与覆盖磁条的磁卡相似。智能卡上的单片机芯片包含 CPU、EPROM、RAM、ROM 和芯片操作系统(Chip Operating System,COS),不仅具有读写和存储数据的功能,而且能对数据进行处理,因此,智能卡被称为最小的个人计算机。目前,智能卡在许多应用领域取代了磁卡。

单用智能卡作为用户的身份凭证仍有不足之处,如果智能卡丢失,那么捡到卡的人就可以假冒真正的用户。因此需要额外的且在智能卡上不具有的信息进行辅助认证,这种信息通常采用个人识别号(Personal Identification Number,PIN)。在验证过程中,验证者不但要验证持卡人的卡是真实的卡,同时还要通过 PIN 码来验证持卡人的确是他本人。采用这种双因素认证机制进一步提升了身份认证的可靠性。

智能卡体积小,方便携带,可以在任何地点进行电子交易,智能卡的读卡器也越来越普遍,有 USB 型的,也有 PC 型的,在 Windows 终端上也可以设置智能卡插槽。

目前在网上银行中采用的高级别安全工具 USB Key 是当前比较流行的智能卡身份认证方式。USB Key 结合了现代密码学技术、智能卡技术和 USB 技术,是新一代身份认证产品,具有以下特点。

(1) 双因素认证。

每一个 USB Key 都具有硬件 PIN 码保护,用户只有同时取得 USB Key 和用户 PIN 码,

才能登录系统。

(2) 带有安全存储空间。

USB Key 具有 8～128KB 的安全数据存储空间，可以存储数字证书、用户密钥等秘密数据，对该空间的读写操作必须通过程序实现，用户无法直接读取，且用户私钥是无法导出的，杜绝了复制用户数字证书或身份信息的可能性。

(3) 硬件实现加密算法。

USB Key 内置 CPU 或者智能卡芯片，可以实现 PKI 体系中使用的数据签名、加解密等各种算法，加解密运算在 USB Key 内进行，保证了用户密钥不会出现在计算机内存中，从而杜绝了用户密钥被黑客截取的可能性。

USB Key 身份认证系统的模式有以下几种：

(1) 基于挑战-应答的双因素认证方式。

先由客户端向服务器发出一个验证请求，服务器接收到此请求后生成一个随机数(挑战)并通过网络传输给客户端，客户端将收到的随机数通过 USB 接口提供给智能卡的计算单元，由计算单元使用该随机数与存储在安全存储空间中的密钥进行运算得到一个结果(应答)作为认证证据传给服务器。与此同时，服务器也使用该随机数与存储在服务器数据库中的该客户密钥进行相同运算，如果服务器的运算结果与客户端回传的响应结果相同，则认为客户端是一个合法用户。这种模式的挑战/应答认证方式只能对客户端的身份进行认证，无法实现对服务端的身份认证。

(2) 基于数字证书的认证方式。

随着 PKI 技术的成熟，许多应用开始使用数字证书进行身份认证与数字加密。数字证书是由权威公正的第三方机构即 CA 中心签发的，以数字证书为核心的加密技术，可以对网络上传输的信息进行加密和解密、数字签名，确保网上传递信息的机密性、完整性，以及交易实体身份的真实性、签名信息的不可否认性，从而保障了网络应用的安全性。USB Key 作为数字证书的存储介质，可以保证数据证书不被复制，并可以实现所有数字证书的功能。

USB Key 中预置了加密算法、摘要算法、密钥生成算法等，可利用密钥生成算法首先为用户生成一对公/私钥，私钥保存在 USB Key 中，公钥可以导出向 CA 申请生成数字证书，数字证书也保存在 USB Key 中。在进行客户端身份认证时，客户端向服务器发送数字证书，服务端利用 CA 的公钥验证数字证书的真实性，完成对客户端身份的认证。客户端可也要求服务端发送数字证书以验证服务端的真实身份。

【例 4-6】 U 盾，即工行 2003 年推出的客户证书 USB Key，是工商银行为客户提供的网上银行业务的高级别安全工具。该产品采用了目前国际领先的信息安全技术，核心硬件模块由 CPU 及加密逻辑、RAM、ROM、EEPROM 和 I/O 5 部分组成，是一个具有安全体系的小型计算机。除了硬件，安全实现完全取决于技术含量极高的智能卡芯片操作系统(COS)，该操作系统就像 DOS、Windows 等操作系统一样，管理着与信息安全密切相关的各种数据、密钥和文件，并控制各种安全服务。从技术角度看，U 盾是用于网上银行电子签名和数字认证的工具，基于 PKI 技术，采用 1024 位非对称密钥算法对网上数据进行加密、解密和数字签名，确保网上交易的保密性、真实性、完整性和不可否认性。USB Key 具有硬件真随机数发生器，密钥完全在硬件内生成，并存储在硬件中，能够保证密钥不出硬件，硬件提供的加解密算法完全在加密硬件内运行。

网上银行用户发起业务指令(例如支付 5000 元)时,被要求插入 USB Key,同时输入与之相对应的 PIN 码进行身份验证,验证成功后,客户端计算机将敏感的网上银行业务指令传输到 USB Key 中,USB Key 中的芯片操作系统先将指令进行 Hash 运算以形成数字摘要,同时用 USB Key 中的私钥对数字摘要加密以产生数字签名。另外,客户端随机产生 DES 密钥,用 DES 密钥对刚才生成的数字签名、网上银行业务指令原文进行加密,同时用服务端的公钥对 DES 密钥加密,将这两部分加密信息通过 USB 接口提交到客户端系统,进而提交给网上银行服务器系统。网上银行服务器方收到客户端信息后,使用自己对应的私钥解密得到 DES 密钥,然后用 DES 密钥解密得到数字签名及指令,之后用客户端公钥验证数字签名,可有效防止指令中途被篡改,并且能够保证用户身份不可抵赖性。客户端 USB Key 的工作流程如图 4.7 所示。

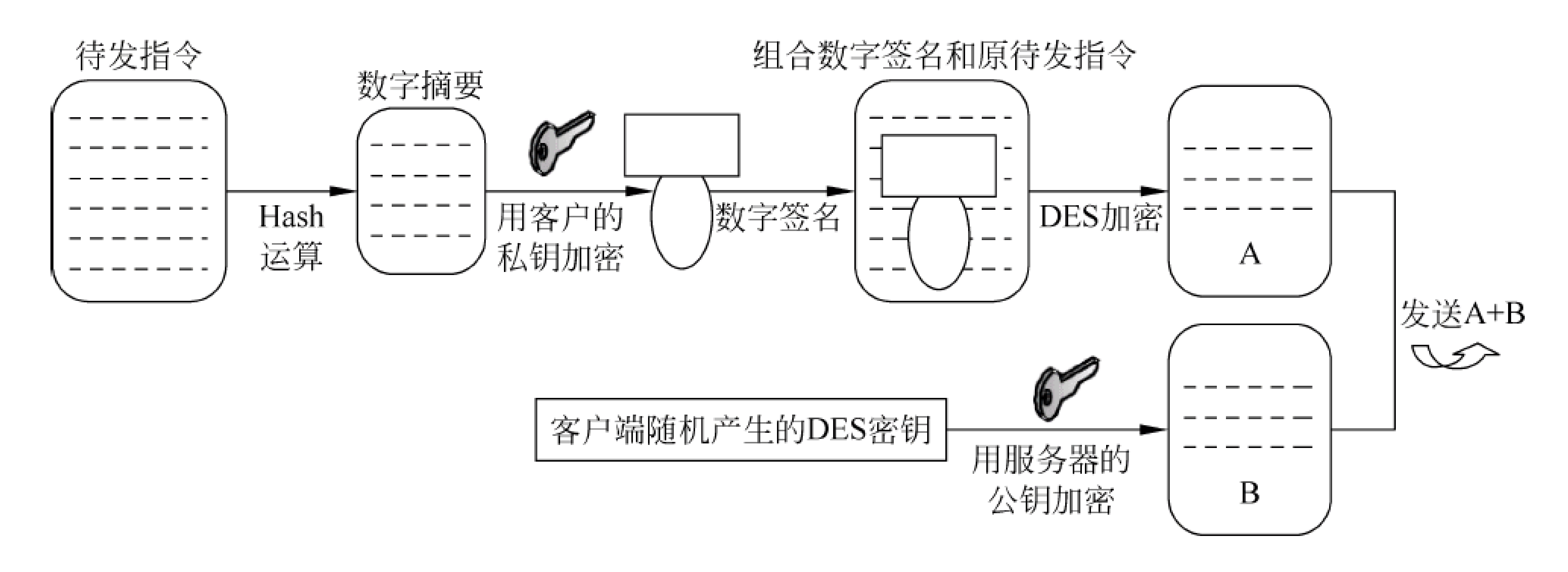


图 4.7　USB key 工作流程

从技术上来讲,USB Key 算得上是目前最为安全的网上银行认证工具。当前几乎所有的网上银行都采取这种基于硬件的数字证书身份认证的方法来保护用户网上银行交易安全。不过由于内置了芯片,USB Key 的制作成本也相对较高,一个功能完备的 USB Key 的市场价格大约是 50～60 元。

利用 USB Key 方式进行身份认证尽管看起来很完美,但实际上还存在一些安全问题。USB Key 认证系统主要存在以下两个安全漏洞:一是黑客完全有能力截获从计算机上输入的静态 PIN 码,用户没有及时取走 USB Key 的时候,黑客便可利用 PIN 码来取得虚假认证,进行转账操作。二是在用户发出交易指令后,通过 USB 接口传送到 USB Key 之前存在一段安全真空期,传输过程中的信息没有受到任何保护,黑客此时可悄无声息地篡改用户指令,而 USB Key 还会坚定地保护篡改后的指令。交易完成后,用户才会发现刚才转账的过程出现了差错,这时损失已经不可挽回。

4.5　基于生物特征的认证方式

传统的用户身份认证机制有许多缺点,目前,虽然从最早的“用户名＋口令”方式过渡到了最新的在网银中广泛使用的 USB Key 方式,但它仍然有许多缺点,首先需要随时携带智能卡,其次它也容易丢失或失窃,补办手续烦琐,并且仍然需要用户出具能够证明身份的其他文件,使用很不方便。直到生物识别技术得到成功应用,身份认证机制才真正回归到了人

类最原始的特性上。基于生物特征的认证技术具有传统的身份认证手段无法比拟的优点。采用生物鉴别技术(Biometrics),可不必再记忆和设置密码,使用更加方便。生物特征鉴别技术已经成为一种公认的、最安全的和最有效的身份认证技术,将成为IT产业最为重要的技术革命。

基于生物特征的身份认证方式是利用人体固有的生理特征或行为特征来进行身份识别或验证。生理特征与生俱来,多为先天性的,如指纹、眼睛虹膜、声音、人脸等,行为特征则是习惯使然,多为后天性的,如笔迹、步态等。这些生物特征具有唯一性、稳定性和难以复制性,采用生物认证技术具有很好的安全性、可靠性和有效性,与传统的身份确认手段相比,具有无法比拟的优点。近几年来,全球的生物识别技术已从研究阶段转向应用阶段,对该技术的研究和应用如火如荼,前景十分广阔。

生物识别的核心在于如何获取这些生物特征,并将之转换为数字信息,存储于计算机中,利用可靠的匹配算法来完成验证和识别个人身份。基于生物特征的认证机制的一般过程如下:

(1) 认证系统先对用户的生物特征进行多次采样,然后对这些采样进行特征提取,并将平均值存放在认证系统的用户数据库中。

(2) 鉴别时,对用户的生物特征进行采样,并对这些采样进行特征提取。通过数据的保密性和完整性保护措施将提取的特征发送到认证系统,并在认证系统上解密用户的特征。

(3) 比较步骤(1)和步骤(2)的特征,如果特征匹配达到近似要求,认证系统向用户返回鉴别成功的信息,否则返回鉴别失败的信息。

与传统身份鉴别相比,生物识别技术具有以下特点:

① 随身性:生物特征是人体固有的特征,与人体是唯一绑定的,具有随身性。

② 安全性:人体特征本身就是个人身份的最好证明,满足更高的安全需求。

③ 唯一性:每个人拥有的生物特征各不相同。

④ 稳定性:生物特征如指纹、虹膜等,不会随时间等变化。

⑤ 广泛性:每个人都具有这种特征。

⑥ 方便性:生物识别技术不需要记忆密码与携带使用特殊工具,不会遗失。

⑦ 可采集性:选择的生物特征易于测量。

基于生物特征的认证系统安全性高,但成本也高。另外,技术上的发展也证明,这些生物特征在安全上并不是无懈可击的。例如有研究者在2002年发现可以用凝胶铸成的指纹模子来瞒骗指纹识别器。同时生物认证系统并不具有普适性,例如人的视网膜图案是独一无二的,用视网膜做认证的确十分可靠,但是由于需要使用聚光灯来获取独特的眼球后面的血管图,并不是每个登录系统的人都愿意用一个仪器来扫描自己的眼睛,因为人的肉眼暴露在这种视网膜扫描装置下会觉得很不舒服,所以这样的系统只在高安全环境中实施。

4.6 身份认证协议

在网络中,通常是服务器要验证用户的身份,称为单向认证(One-Way Authentication),但是为了防止网络钓鱼等假冒服务器的现象发生,客户端有时也需要验证服务器的身份,这

就需要进行双向认证(Two-Way Authentication),下面依次介绍采用挑战/应答方式的认证过程。

4.6.1 单向认证

单向认证是通信的一方认证另一方的身份,例如服务器在向用户提供服务之前,先要认证用户是否是这项服务的合法用户,但是不需要向用户证明自己的身份,单向认证可采取普通的口令认证,也可用对称密码或非对称密码体制实现。

1. 采用对称密码体制实现单向认证

假设 Bob 要验证 Alice 的身份,Alice 首先向 Bob 提出认证请求,Bob 产生一个随机数 R_B 发送给 Alice,Alice 用双方共享的密码 $K_{A\text{-}B}$ 加密该随机数后发送给 Bob,Bob 如果能用相同的密码解密得到 R_B,则 Alice 的身份得到 Bob 的验证,因为密码 $K_{A\text{-}B}$ 是 Alice 和 Bob 共享的秘密,其他任何人没有该密钥就无法构造出 R_B 的密文,如图 4.8 所示。

采用对称密码体制进行双单向身份认证的方式有很多,上例只是其中的一种。

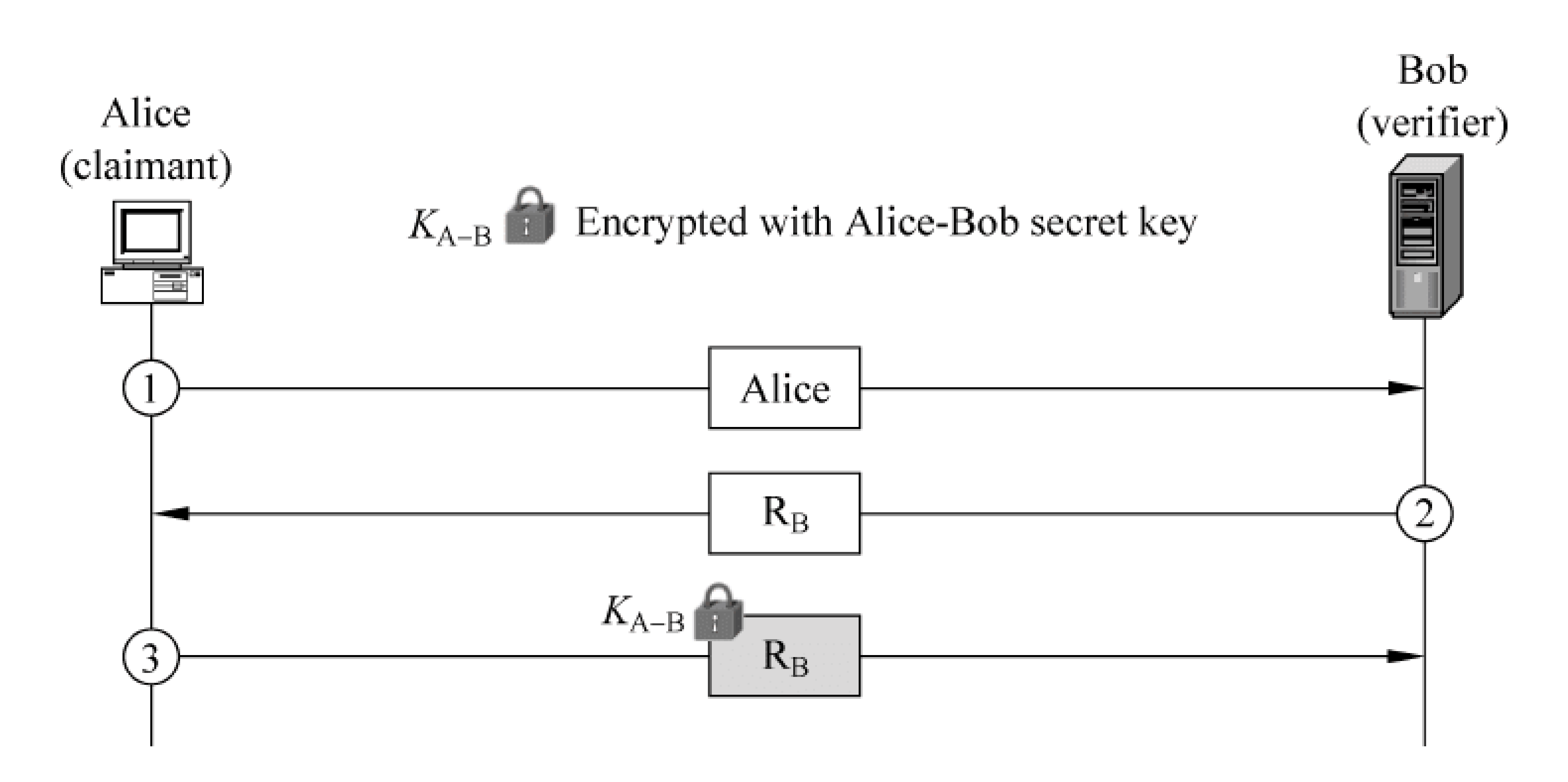


图 4.8 采用对称密码体制实现单向认证

2. 采用非对称密码体制实现单向认证

这里 Bob 要验证 Alice 的身份,Alice 首先向 Bob 提出认证请求,Bob 产生一个随机数 R_B 并用 Alice 的公钥加密后发送给 Alice,Alice 用私钥解密得到随机数后发送给 Bob,Bob 验证发送过来的随机数,则 Alice 的身份得到 Bob 的验证,如图 4.9 所示。

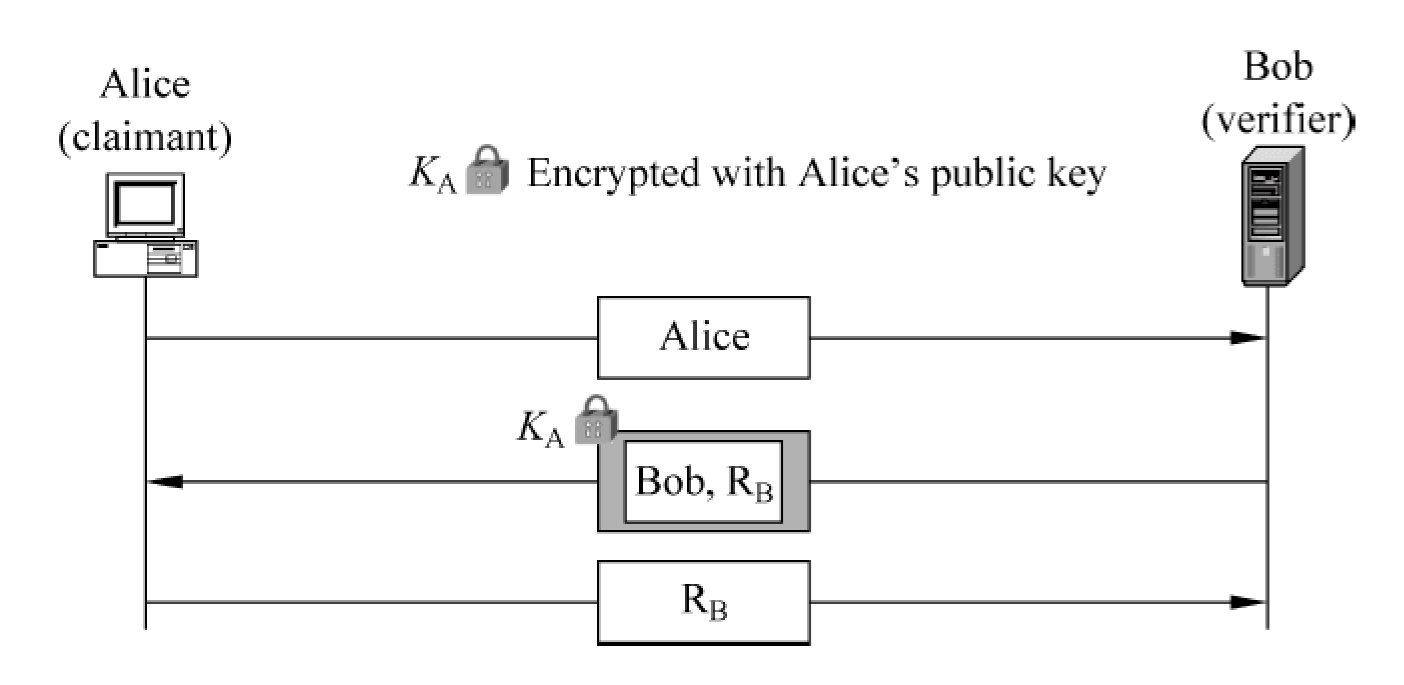


图 4.9 采用公钥密码体制实现单向认证

采用公钥密码体制进行单向身份认证的方式有很多，上例只是其中的一种。

4.6.2 双向认证

双向认证需要通信双方互相认证对方的身份，同样可以采用普通的口令认证方式，这里主要讨论采用对称密码体制和非对称密码体制实现双向身份认证。

1. 用对称密码体制实现双向认证

Alice 首先向 Bob 提出认证请求，Bob 产生一个随机数 R_B 发送给 Alice，Alice 为了同时验证 Bob 的身份，也产生一个随机数 R_A，串接在 R_B 后并用双方共享的密钥 $K_{A\text{-}B}$ 加密，密文发送给 Bob，Bob 由于拥有密钥 $K_{A\text{-}B}$，可以解密，如能分解出 R_B，则验证通过 Alice 的身份，同时为了验证自己的身份(也即拥有密钥 $K_{A\text{-}B}$)，他将 R_A、R_B 调换顺序串接后用共享密钥加密后发送给 Alice，Alice 解密后如能分离出 R_A，则验证通过 Bob 的身份，如图 4.10 所示。

采用对称密码体制进行双向身份认证的方式有很多，上例只是其中的一种。

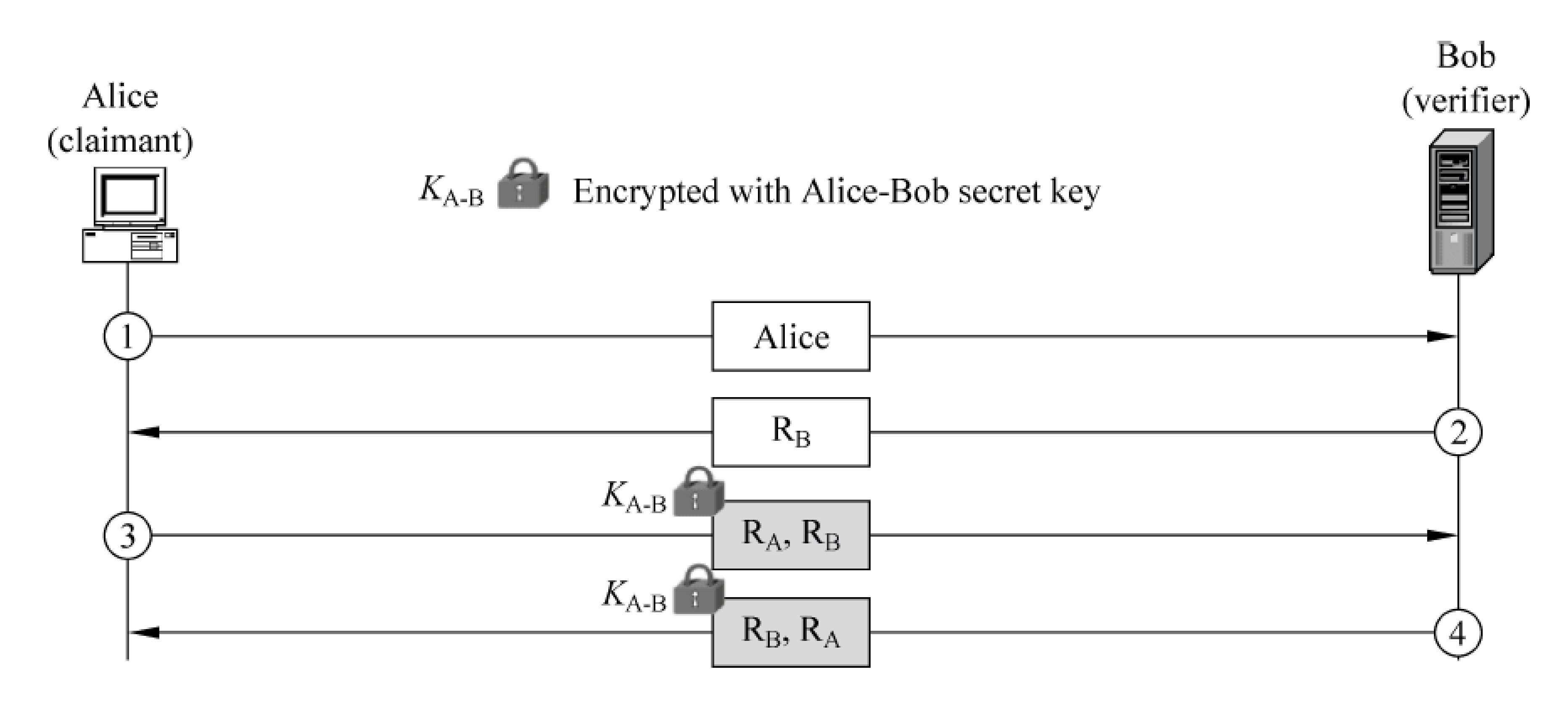


图 4.10 采用对称密码体制实现双向认证

2. 用公钥密码体制实现双向认证

Alice 首先产生一个随机数 R_A，用 Bob 的公钥加密后发送给 Bob，Bob 接收到后用自己的私钥解密得到 R_A，产生一个随机数 R_B 串接在 R_A 后用 Alice 的公钥加密后发送给 Alice，Alice 用自己的私钥解密，如果能分离出 R_A 则验证通过 Bob 的身份，同时为了验证自己的身份将解密分离出的 R_B 发送给 Bob，Bob 接收到之后可以验证 Alice 的身份，如图 4.11 所示。

采用公钥密码体制进行双向身份认证的方式有很多，上例只是其中的一种。

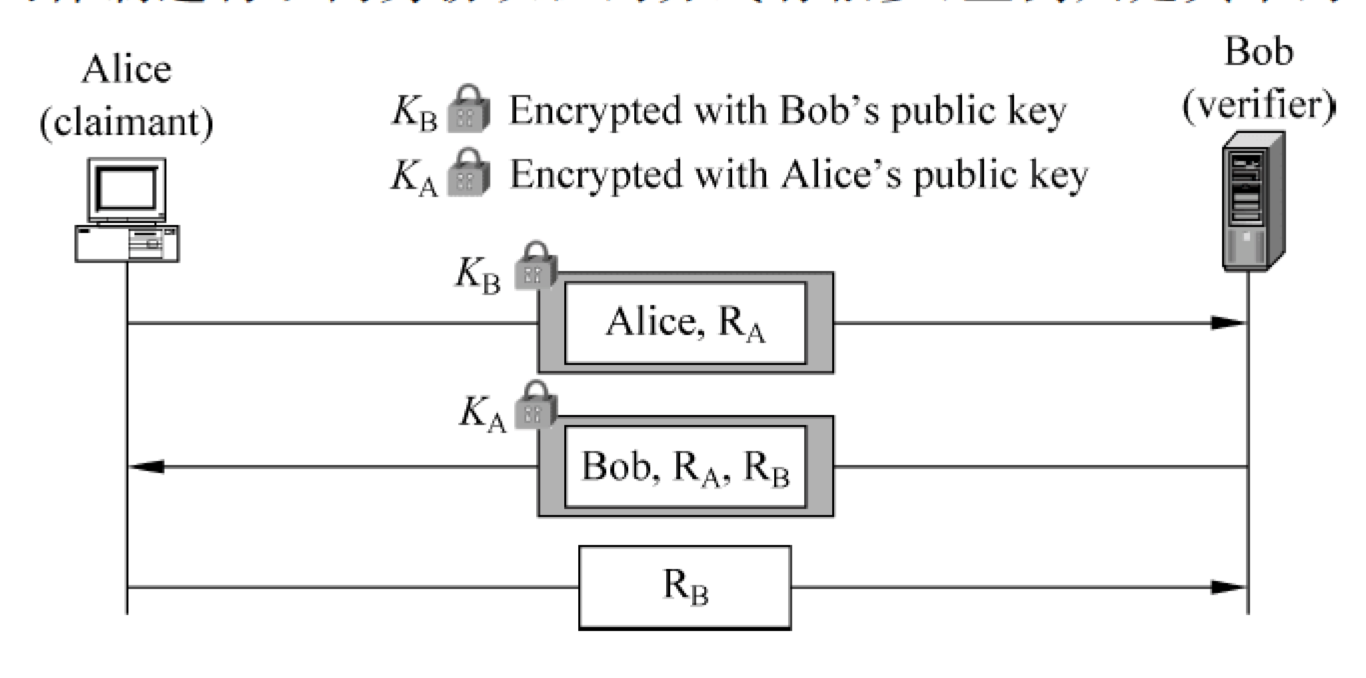


图 4.11 采用公钥密码体制实现双向认证

4.6.3 可信的第三方认证

可信的第三方(Trusted Third Party)认证也是一种通信双方相互认证的方式,但是认证过程必须借助于一个双方都能信任的第三方,一般而言可以是政府机构或其他可信赖的机构。当两端欲进行通信时,彼此必须先通过信任的第三方认证,然后才能互相交换密钥,而后进行通信。

由借助于信任第三方的认证方式变化而来的认证协议相当多,各有各的特色与优缺点,其中一个最著名的例子就是由美国麻省理工学院提出的 Kerberos 协议。Kerberos 是在 20 世纪 80 年代中期作为美国麻省理工学院“雅典娜计划”(Project Athena)的一部分开发的,有关 Kerberos 协议的具体内容请参考网络安全协议相关书籍。

4.7 零知识证明

通常的身份认证都要求传输口令或身份信息,如果不透露这些信息,身份也能得到证明就好了,这就需要零知识证明技术(the Proof of Zero Knowledge)。

这里假设 P 为示证者,V 为验证者,P 试图向 V 证明自己知道某消息,一种方法是 P 说出这一消息使 V 相信,这样 V 也知道了这一消息,这是基于知识的证明;另一种方法是使用某种有效的数学方法,使得 V 相信他掌握这一信息,却不泄露任何有用的信息,这种方法称为零知识证明。

解释零知识证明概念的最经典的例子是 1990 年 Louis C. Guillou 和 Jean-Jacques Quisquater 提出的“洞穴问题”,如图 4.12 所示,洞穴里面有一个秘密咒语,只有知道咒语的那些人才能打开 C 和 D 之间的密门。C 与 D 之间的横线为一扇门,P 知道打开这扇门的咒语,现在 P 要向 V 证明他知道这个咒语,按照传统的做法是 P 把这个咒语告诉 V,然后 V 用这个咒语去打开那扇门,如果能打开说明这个咒语是正确的,那么就可以验证 P 的身份。这样做虽然成功地证明了 P 的身份,但是也泄露了这个咒语,那么 V 可以冒充 P 或者 V 可以把这个咒语透露给第三者,从而使得这个系统存在安全隐患。

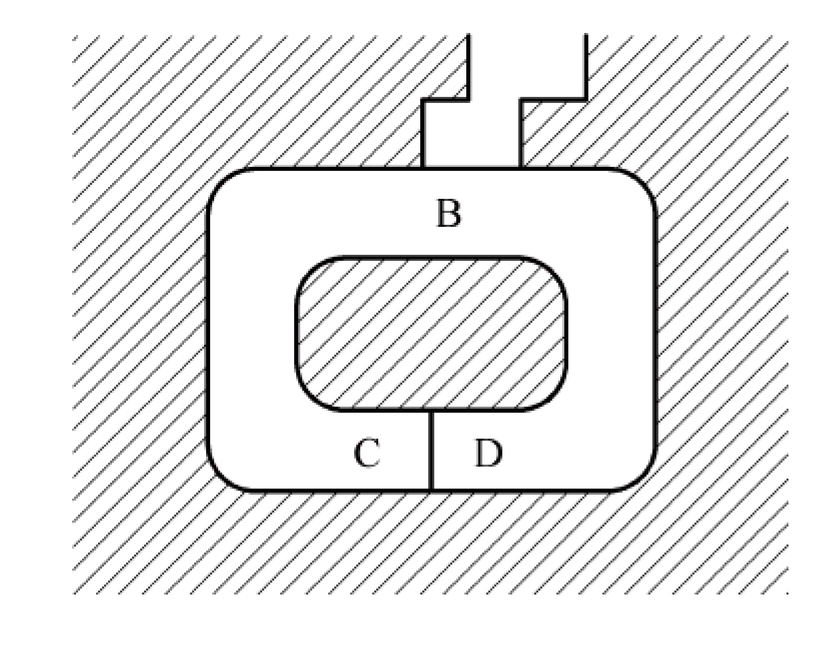


图 4.12 洞穴问题

以下是用零知识证明理论来证明 P 的身份。

(1) V 站在 A 点,让 P 进入洞内,这样保证 V 不知道 P 是从哪边进入洞内的(从左边进入到达 C 点,从右边进入则到达 D 点)。

(2) 当 P 到达 C 点或 D 点以后,V 走到 B 点。

(3) V 向 P 喊叫,要他从左通道或右通道出来。

(4) P 按照 V 的要求从 V 指定的方向出来,如果有必要他就用咒语打开密门。

(5) P 和 V 重复步骤(1)~(4)n 次。

若 P 不知咒语,只有 1/2 的机会猜中 V 的要求,协议执行 n 次,则只有 2^{-n}的机会完全

猜中,若 $n=16$,则若每次均通过 V 的检验,则 V 受骗的机会仅为 1/65 636。如果 n 足够大,几乎就可以证明 P 是知道这个咒语的并且 V 并没有得到任何关于这个咒语的信息。

最早出现的零知识身份证明协议是 Feige-Fiat-Shamir 协议(简称 FFS 协议),在 Feige-Fiat-Shamir 协议中,可信赖第三方选定 $m=p\times q$,其中 p、q 为两个大素数,m 为 512bit 或者为 1024b。通信双方共享 m,再由可信赖第三方实施公钥私钥的分配,产生 k 个随机数 $v_1,v_2,\cdots,v_k$(要求 $v_i^{-1} \bmod m$ 存在,$i=1,2,\cdots,k$),且使 v_i 为模 m 的平方剩余,即 $x^2=v_i \pmod m$,v_i 为公钥,计算 $s_i=\sqrt{v_i^{-1}} \bmod m$,$s_i$ 作为示证方的私钥(这里 P 作为示证方,V 作为验证方)。

身份验证协议如下。

(1) 用户 P 取随机数 $r(r<m)$,计算 $x=(r^2) \bmod m$,将 x 发送给 V。

(2) V 发送挑战信息串 $b_1,b_2\cdots,b_k$(b_i 为 0 或者 1)给 P。

(3) P 接受挑战,算出 $y=r\times\prod_{i=1}^{k} s_i^{bi} \bmod m$ 作为应答发送给 V。

(4) V 验证 P 的身份,验证 $x=y^2\prod_{i=1}^{k} vi^{bi} \bmod m$,如果相等则受骗概率为 2^{-k}。如果不相等,这个验证过程则终止。

P 和 V 可以重复执行此协议,每次以不同的 r 和 b 串,执行完一次验证过程后 P 能欺骗 V 的概率为 2^{-k}。

Feige-Fiat-Shamir 协议需要经过多次交互才能保证身份认证的安全性,不仅费时而浪费系统资源,随后又提出了其他的基于零知识证明的身份认证协议,这里不再详细讨论,零知识证明的出现给身份认证带来了一个新方向。

4.8 小　　结

身份认证是验证主体的真实身份与其所声称的身份是否相符的过程,它是信息系统的第一道安全防线。身份认证方式主要有三种:利用用户所知道的、用户所拥有的以及利用用户的生物特征,三种身份认证方式都有自身的弱点,即口令可能会被遗忘,智能卡可能会被偷盗,生物特征的实现代价高、识别率相对较低。为了取得更高的安全性,综合运用这三种认证方式的认证机制称为双因素认证。双因素认证已经得到了广泛的应用,例如 USB Key 等。

习　　题

一、填空题

1. 身份认证包括________、________两个过程。
2. ________是最常见的身份认证方式。
3. 针对弱口令的攻击主要有________,________。

4. ________用来抵御重放攻击。

5. 网上银行中采用的身份认证方式有________,________。

6. 不泄漏秘密信息进行身份认证的技术称为________。

二、选择题

1. Windows NT 和 Windows 2000 系统能设置为在几次无效登录后锁定账号,这可以防止(　　)。

A. 木马　　B. 暴力攻击　　C. IP 欺骗　　D. 缓存溢出攻击

2. 在以下认证方式中,最常用的认证方式是(　　)。

A. 基于账户名/口令认证　　B. 基于摘要算法认证

C. 基于 PKI 认证　　D. 基于数据库认证

3. 以下哪项不属于防止口令猜测的措施?(　　)

A. 严格限定从一个给定的终端进行非法认证的次数

B. 确保口令不在终端上再现

C. 防止用户使用太短的口令

D. 使用机器产生的口令

三、简答题

1. 什么是字典攻击和重放攻击?什么是一次性口令认证?为什么口令加密过程要加入不确定因子?

2. 单机状态下验证用户身份的三种因素是什么?

3. 一次性口令认证实现技术有哪些?

4. 简述基于 PKI 技术体系的 USB Key 认证原理。

5. 自行设计一个利用公钥密码体制实现双向认证的协议。

第 5 章　访 问 控 制

访问控制是信息安全保障机制的重要内容,它是实现数据保密性和完整性机制的主要手段之一。访问控制是在身份认证的基础上,根据身份对提出的资源访问请求加以控制,其目的是为了保证网络资源受控、合法地使用,用户只能根据自己的权限来访问系统资源,不能越权访问,同时,访问控制也是记账、审计的前提。广义地讲,所有的计算机安全都与访问控制有关。

本章 5.1 节介绍了访问控制的概念和组成要素;5.2 节具体介绍三种访问控制机制,即自主访问控制、强制访问控制和基于角色的访问控制机制,详细介绍了自主访问控制的三种实现方式以及强制访问控制的安全模型。

5.1　访问控制概述

5.1.1　访问控制机制与系统安全模型

James P. Anderson 在 1972 年提出的引用监控器(The Reference Monitor)的概念是经典安全模型的最初雏形,如图 5.1 所示。

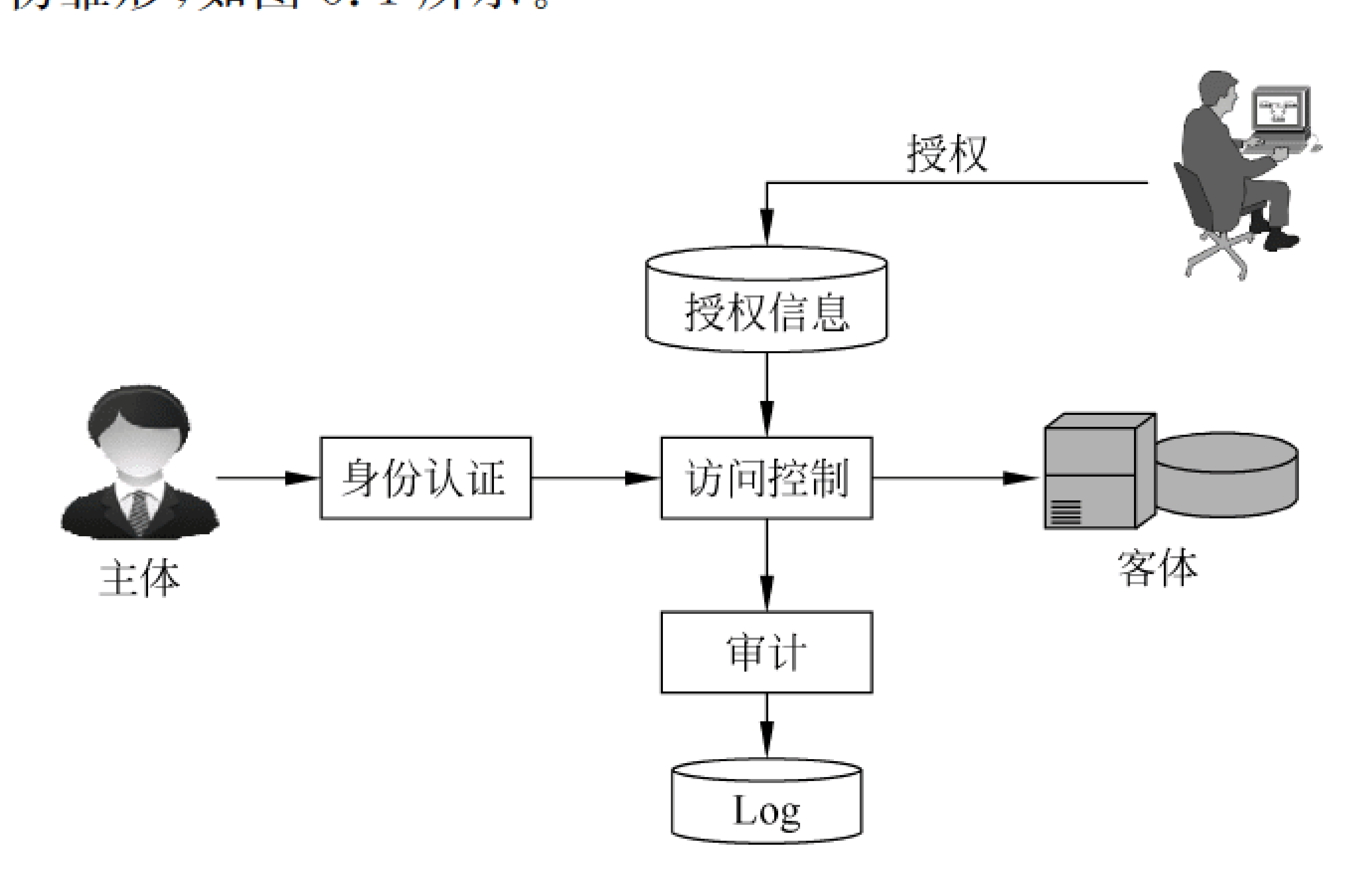


图 5.1　引用监控器模型

经典安全模型包括如下基本要素:

(1) 明确定义的主体和客体;

(2) 描述主体如何访问客体的一个授权数据库;

(3) 约束主体对客体访问尝试的引用监控器;

(4) 识别和验证主体和客体的可信子系统;

(5) 审计引用监控器活动的审计子系统。

从图 5.1 中可以看出,实现计算机系统安全的基本措施(安全机制)包括身份认证(识别和验证)、访问控制和审计。身份认证是验证用户的身份与其所声称的身份是否一致的过

程。访问控制是在主体身份得到认证后，根据授权数据库中预先定义的安全策略对主体行为进行限制的机制和手段。审计作为一种安全机制，它在主体访问客体的整个过程中都发挥作用，为安全分析提供了有力的证据支持。本章主要讨论访问控制技术。

5.1.2　访问控制的基本概念

访问控制技术起源于 20 世纪 70 年代，当时是为了满足管理大型主机系统上共享数据授权访问的需要。随着计算机和网络技术的发展，访问控制技术在信息系统的各个领域得到了越来越广泛的应用，先后出现了多种重要的访问控制技术，如自主访问控制、强制访问控制、基于角色的访问控制等。

访问控制常常以身份认证作为前提，在此基础上实施各种访问控制策略来控制和规范合法用户在系统中的行为，身份认证解决的是“你是谁，你是否真的是你所声称的身份”，目的是阻止非法用户进入系统，而访问控制技术解决的是“你能做什么，你有什么样的权限”，目的是限制合法用户的操作权限。

访问控制包括两个重要的过程，其一是系统通过授权（Authorization）设定合法用户对资源的访问权限规则集；其二是根据预先设定的规则对用户访问某项资源（目标）的行为进行控制，只有规则允许时才能访问，违反预定安全规则的访问行为将被拒绝。资源可以是信息资源、处理资源、通信资源或者物理资源，访问方式可以是获取信息、修改信息或者完成某种功能，一般情况可以理解为读、写或者执行。

访问控制是针对越权使用资源的防御措施，通过限制对关键资源的访问，防止非法用户的侵入或因为合法用户的不慎操作而造成的破坏，从而保证网络资源受控、合法地使用。访问控制中涉及的主要概念包括以下几个：

1. 主体（Subject）

主体是指访问操作的主动发起者，它造成了信息的流动和系统状态的改变，主体可以是用户或其他任何代理用户行为的实体，如进程、作业等。

2. 客体（Object）

客体是被访问的对象，客体在信息流动中的地位是被动的，处于主体作用之下。凡是可以被操作的对象都可以认为是客体，客体通常包括文件、目录、消息、程序、库表等，还可以是处理器、通信信道、时钟、网络节点等。

3. 访问（Access）

访问是使信息在主体和客体之间流动的一种交互方式。访问包括读取数据、更新数据、运行程序、发起连接等。

4. 访问控制策略

访问控制策略是主体对客体的访问规则集，这个规则集直接定义了主体对客体的作用行为和客体对主体的条件约束。访问策略体现了一种授权行为，也就是客体对主体的权限允许，这种允许不超越规则集中的定义。

访问控制策略的制定需要考虑以下原则：

1）最小特权原则

最小特权原则是指主体执行操作时，按照主体所需权利的最小化原则分配给主体权利。

最小特权原则的优点是最大限度地限制主体行为，可以避免来自突发事件、错误和未授权主体的危险。也就是说，为了达到一定目的，主体必须执行一定操作，但他只能做他所被允许做的。

2）最小泄露原则

最小泄露原则是指主体在执行任务时，按照主体所需知道的信息最小化的原则分配给主体权利。

3）多级安全策略

主客体分配一定的安全级别，安全级别通常包括绝密、秘密、机密、限制和无级别 5 级。主客体的数据流向和权限控制以不允许信息从高级别向低级别流动为原则，采用多级安全策略可以避免敏感信息的扩散。

访问控制在信息系统中的应用非常广泛，例如对用户的网络接入过程进行控制、操作系统中控制用户对文件系统和底层设备的访问。另外当需要提供更细粒度的数据访问控制时，可以在应用程序中实现基于数据记录或更小的数据单元访问控制。例如大多数数据库管理系统（如 Oracle）都提供独立于操作系统的访问控制机制，Oracle 使用其内部用户数据库，且数据库中的每个表都有自己的访问控制策略来支配对其记录的访问。

5.2 访问控制策略

1985 年美国军方提出了可信计算机系统评估准则 TCSEC，其中描述了两种著名的访问控制策略：自主访问控制和强制访问控制。基于角色的访问控制（RBAC）由 Ferraiolo 和 Kuhn 在 1992 年提出，考虑到网络安全和传输流，又提出了基于对象和基于任务的访问控制。

各种访问控制策略之间并不相互排斥，现存计算机系统中通常都是多种访问控制策略并存，系统管理员能够对安全策略进行配置使其达到安全政策的要求。

5.2.1 自主访问控制

自主访问控制（Discretionary Access Control，DAC）是指资源的所有者（往往是创建者），对于其拥有的资源，可以自主地将访问权限分发给其他主体，即确定这些主体对于资源有怎样的访问权限，是最常用的访问控制机制。在这种访问控制机制下，客体的拥有者可以按照自己的意愿精确指定系统中其他用户对其客体的访问权，从这种意义上来说，是“自主的”。Linux、UNIX、Windows NT/SERVER 版本的操作系统，SQL Server、Oracle 等数据库管理系统都提供了自主访问控制的功能。自主访问控制通常有三种实现机制，即访问控制矩阵（Access Control Matrix）、访问控制列表（Access Control Lists，ACLs）和访问控制能力表（Access Control Capabilities Lists，ACCLs）。

1. 访问控制矩阵

访问控制矩阵是最初实现访问控制机制的概念模型，它利用二维矩阵规定了任意主体和任意客体间的访问权限。矩阵中的行代表主体的访问权限属性，矩阵中的列代表客体的访问权限属性，矩阵中的每一格表示所在行的主体对所在列的客体的访问授权。如表 5.1 所示，其中 Own 表示所在行主体是所在列客体的属主，可以自主授予或回收其他用户对其

拥有客体的访问权限,即拥有对客体管理的权限,R 表示读操作,W 表示写操作。

表 5.1　访问控制矩阵示例

	File1	File2	File3	File4
张三	Own,R,W		Own,R,W	
李四	R	Own,R,W	W	R
王五	R,W	R		Own,R,W

访问控制矩阵清晰地描述了任意主体对任意客体的访问权限,但是,在较大的系统中,访问控制矩阵将变得非常巨大,而且矩阵中的许多任务格可能为空,造成很大的存储空间浪费,因此在实际应用中,访问控制很少利用矩阵方式实现,目前大部分系统实现的自主访问控制是用基于访问控制矩阵的行或列来表达访问控制信息。

2. 访问控制列表

访问控制列表实际上是按访问控制矩阵的列实施对系统中客体的访问控制,是从客体角度进行设置的、面向客体的访问控制。每个客体有一个访问控制列表,用来说明有权访问该客体的所有主体及访问权限,如图 5.2 所示。利用访问控制列表,能够很容易地判断出对于特定客体的授权访问。

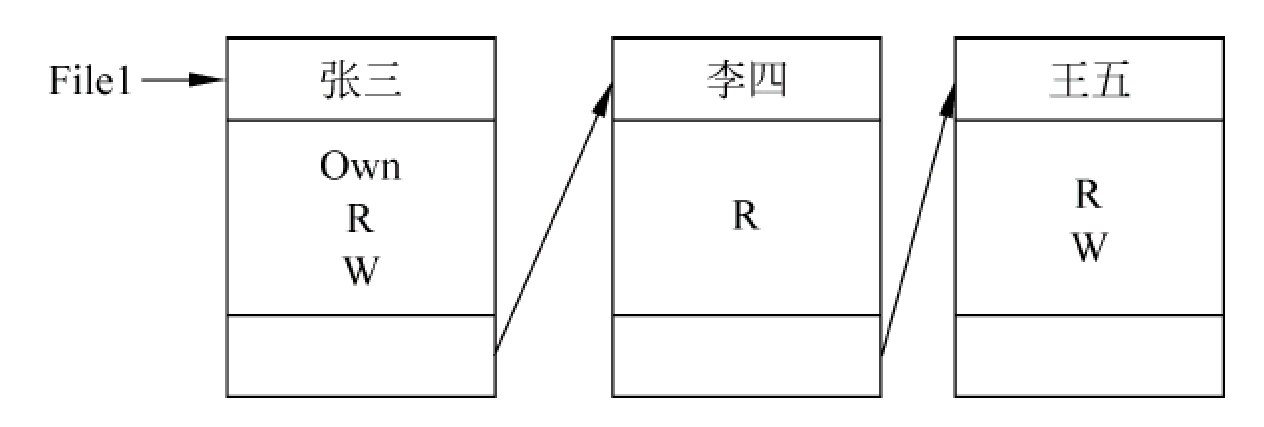


图 5.2　访问控制列表

由于访问控制列表简单、实用,虽然在查询特定主体能够访问的客体时,需要遍历查询所有客体的访问控制列表,它仍然是一种成熟且有效的访问控制实现方法,许多通用的操作系统使用访问控制表来提供访问控制服务。

【例 5-1】 Linux 中实现了访问控制列表的简略方式,将系统中的所有用户划分为三类:属主用户、同组用户、其他用户,系统按这三类用户进行授权,权限主要包括 r:读,w:写,x:执行,这样可以使得访问控制列表只需要 9 位就可描述。如某个文件的访问控制列表为“rwxr-x---”,从左往右每三位为一组,第一组“rwx”表示文件的属主拥有可读、可写、可执行权限,第二组“r-x”表示文件属主的同组用户拥有读和执行权限,第三组“---”表示其他用户对该文件没有访问权限。

3. 访问能力表

访问能力表(Access Capabilities List)实际上是按访问控制矩阵的行实施对系统中客体的访问控制,如图 5.3 所示。能力(Capability)是为主体提供的、对客体具有特定访问权限的不可伪造的标志,它决定主体是否可以访问客体以及以什么方式(如读、写、修改或运行)访问客体。主体可以将能力转移给为自己工作的进程,在进程运行期间,还可以动态地添加或修改能力。能力的转移不受任何策略的限制,所以对于一个特定的客体,不能确定所有有权访问它的主体,利用访问能力表实现自主访问控制的系统并不多。

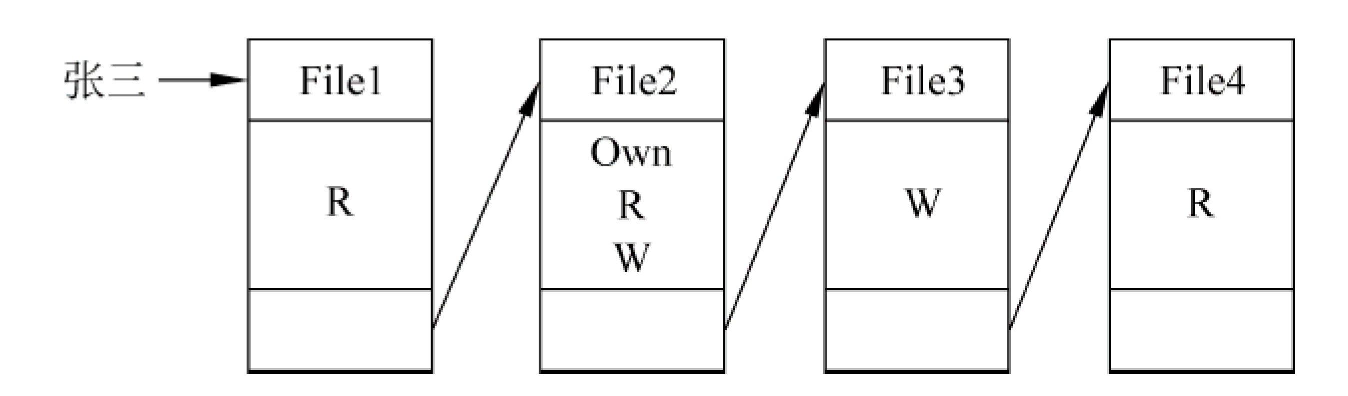


图 5.3　访问控制能力表

自主访问控制的最大特点是自主,即资源的拥有者对其资源的访问策略具有决策权,因此是一种限制比较弱的访问控制策略,这种方式给用户带来灵活性的同时,也带来了安全隐患。这种机制允许用户自主地将自己客体的访问操作权转授给别的客体,权力多次转授后,一旦转授给不可信主体,那么客体的信息就会泄露。DAC 的另一个缺点是无法抵御特洛伊木马的攻击,木马窃取敏感文件的方法有两种,一是通过修改敏感文件的访问权限来获取敏感信息,在 DAC 机制下,某一合法用户可以任意运行一段程序修改自己文件的访问权限,系统无法区分这是合法用户的修改还是木马程序的非法修改;二是躲在用户程序中的木马利用合法用户身份读敏感文件的机会,把所访问文件的内容复制到入侵者的临时目录下,而 DAC 无法阻止,因而无法抵挡特洛伊木马的攻击。

【例 5-2】 假设用户 SOS 将其重要信息存放在文件 important. doc 中,并且将文件权限设置成只有自己可以读写。SPY 是一个恶意攻击者,试图读取 important. doc 文件的内容,他首先准备好一个文件 pocket. doc,并将其权限设置成为 SOS:w,SPY:rw,同时设计一个有用的程序 use_it_please,该程序除了有用部分,还包含一个木马。当诱使 SOS 下载并运行该程序时,木马会以 SOS 用户的身份执行,将 important. doc 中的信息写入 pocket. doc 文件,这样 SPY 就窃取了 important. doc 的内容。

对安全性要求更高的系统,仅采用 DAC 是不够的,需要采用更安全的访问控制技术——强制访问控制。

5.2.2　强制访问控制

1. 强制访问控制的概念

强制访问控制(Mandatory Access Control,MAC)是比 DAC 更为严格的访问控制策略,最早出现在 20 世纪 70 年代,是美国政府和军方源于对信息保密性的要求以及防止特洛伊木马攻击而研发的。

与 DAC 相比,强制访问控制不再让众多的普通用户完全管理授权,而是将授权归于系统管理,并确保授权状态的变化始终处于系统的控制下。在强制访问控制中,每个主体(进程)和客体(文件、消息对列、共享存储区等)都被赋予一定的安全属性,并且安全属性只能由管理部门(如安全管理员)或操作系统按照严格的规则进行设置,当一个进程访问一个客体(如文件)时,强制访问控制机制通过比较进程的安全属性和文件的安全属性来决定访问是否允许,如果系统判定拥有某一安全属性的主体不能访问某个客体,那么即使是客体的拥有者都不能使该主体有权访问客体。MAC 主要用于保护敏感数据(例如,政府、军队敏感文件等)。

在系统中实现 MAC 时,需要根据总体安全策略和需求为系统中的每个主体和客体分

配一个适当的安全级别,且安全级别是不能轻易改变的,它由管理部门(如安全管理员)或由操作系统自动按照严格的规则设置。在 MAC 下,即使是客体的拥有者,也没有对自己客体的控制权,并且系统安全管理员修改、授予、撤销主体对客体的访问权的管理工作,也要受到严格的审核与监控。

2. 强制访问控制模型

1) BLP 模型

1973 年,David Bell 和 Len Lapadula 提出了第一个也是最著名的安全策略模型 Bell-LaPadula 安全模型,简称 BLP 模型,BLP 模型是遵守军事安全策略的多级安全模型,主要用于解决面向机密性的访问控制问题,已实际应用于许多安全操作系统的开发中。

在 BLP 模型中主客体的安全属性由两部分构成。

(1) 保密级别(又称为敏感级别或级别):例如公开、秘密、机密、绝密等。

(2) 一个或多个范畴:该安全级涉及的领域,例如陆军、海军、空军等。

因此一个安全属性包括一个保密级别、一个范畴集,而范畴集包含任意多个范畴,安全属性通常写作保密级后随一个范畴集的形式,例如{机密;陆军,海军,空军}。

在安全属性中,保密级别是线性排列的,例如公开<秘密<机密<绝密,范畴则是相互独立和无序的,两个范畴集之间的关系是包含、被包含或无关。

BLP 模型有两个基本规则。

(1) 规则 1(简单安全性):一个主体对客体进行读操作的必要条件是主体的安全级支配客体的安全级,即主体的保密级别不小于客体的保密级别,主体的范畴集包含客体的全部范畴,即主体只能向下读。

(2) 规则 2(*特性):一个主体对客体进行写访问的必要条件是客体的安全级支配主体的安全级,即客体的保密级别不小于主体的保密级别,客体的范畴集包含主体的全部范畴,即主体只能向上写。

BLP 模型的强制访问控制可以概括为不允许"上读,下写",这种规则是由信息的保密性的安全要求决定的。保密性要求只有高保密级的主体能够读低保密级客体的内容,否则会造成高保密级的客体的信息泄密;反过来,高保密级的主体对低保密级的客体进行写操作也会造成信息泄密,如图 5.4 所示。

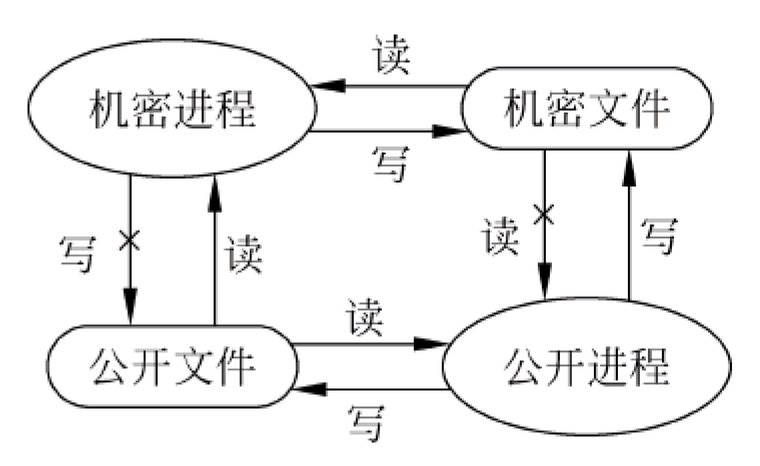


图 5.4 多级安全规则

【例 5-3】 客体 LOGISTRIC 文件的敏感标签为 SECRET[VENUS ALPHA],主体 Jane 的敏感标签为 SECRET[ALPHA],虽然主体的敏感等级满足上述读写规则,但是由于主体 Jane 的类集合当中没有 VENUS,所以不能读此文件,而写则允许,因为客体 LOGISTIC 的敏感等级不低于主体 Jane 的敏感等级,写了以后不会降低敏感等级。

运用 BLP 模型的*特性可有效防范特洛伊木马。前面介绍过木马窃取敏感文件的方法有两种,一是通过修改敏感文件的访问权限来获取敏感信息,在 DAC 机制下,某一合法用户可以任意运行一段程序修改自己文件的访问权限,系统无法区分这是合法用户的修改还是木马程序的非法修改,但在 MAC 下,杜绝了用户修改客体安全属性的可能,因此木马利用这种方法窃取敏感信息是不可能的;二是特洛伊木马伪装成正常的程序,例如一个小

游戏、一个小工具,诱使用户下载运行,而实际当运行带有木马的程序时,木马会利用合法用户的身份读取敏感信息,把所访问的文件复制到入侵者的临时目录下,这在 DAC 机制下是完全可以做到的,然而在 * 特性下,能阻止正在机密安全级上运行的木马,把机密信息写到一个低安全级别的文件中,因为机密级进程写的每条消息的安全级至少是机密级的。

基于 BLP 模型的 MAC 阻止了信息由高级别的主/客体流向低级别的主/客体,保证了信息的机密性,适用于保密性要求比较高的军事、政府部门和金融等领域,但该模型不能保证信息的完整性。而在商业领域,以加强数据完整性为目的的强制访问控制模型也有广泛的应用。

2) Biba 模型

对于政府发布的公告,允许所有用户阅读,但绝对不允许被篡改,在这种应用场合下,更强调数据的完整性保护。

Biba 模型是 BLP 模型的变体,由 Biba 等人于 1977 年提出,它的主要目的是保护数据的完整性。

在 Biba 模型中,每个主体和客体都被分配一个完整性属性,类似于 BLP 模型,该完整性属性是由一个完整性级别和一个范畴集构成的。Biba 模型规定,信息只能从高完整性等级向低完整性等级流动,就是要防止低完整性的信息"污染"高完整性的信息。

Biba 模型并未约定具体采用的策略,而是将策略分为非自主策略和自主策略两类,在每类下给出了一些具体的策略以适应不同的需求,下面简单介绍非自主策略。

非自主策略是指主体是否具有对客体的访问权限取决于主体和客体的完整性级别,具体规则为:

主体对客体进行读访问的必要条件是客体的完整级不低于主体的完整级,即主体只能向上读。

主体对客体进行写操作的必要条件是主体的安全级不低于客体的安全级,即主体只能向下写。

3) Dion 模型

Dion 于 1981 年提出了同时面向机密性和完整性的 Dion 模型,该模型结合 BLP 模型中保护数据机密性的策略和 Biba 模型中保护数据完整性的策略,模型中的每一个客体和主体被赋予一个安全级别和完整性级别,安全级别定义同 BLP 模型,完整性级别定义如 Biba 模型,因此可以有效地保护数据的机密性和完整性。

强制访问控制是比自主访问控制功能更强的访问控制机制,但是这种机制也给合法用户带来许多不便。例如,在用户共享数据方面不灵活且受到限制。因此,当敏感数据需在多种环境下受到保护时,就需要使用 MAC,如需对用户提供灵活的保护且更多地考虑共享信息时,则使用 DAC。

在高安全级(TCSEC 标准的 B 级)以上的计算机系统中常常将自主访问控制和强制访问控制结合在一起使用。自主访问控制作为基础的、常用的控制手段;强制访问控制作为增强的、更加严格的控制手段。一些客体可以通过自主访问控制保护,重要客体必须通过强制访问控制保护。对于通用型操作系统,从用户友好性出发,一般还是以 DAC 机制为主,适当增加 MAC 控制,目前流行的操作系统(如 Windows、UNIX、Linux)、数据库管理系统(SQL Server、Oracle)均属于这种情况。

5.2.3　基于角色的访问控制

1. 概述

MAC 和 DAC 属于传统的访问控制模型，通常为每个用户赋予对客体的访问权限规则集，如果系统中用户数量众多，且系统安全需求处于不断变化中，就需要进行大量烦琐的授权操作，系统管理员的工作将变得非常繁重，更主要的是容易发生错误，造成安全漏洞。

在现实的工作中，绝大多数情况并不是针对每个人设定其工作职责，而是根据这个人在工作单位中所承担的角色设定其工作职责的，例如医院包括医生、护士、药剂师等角色，而银行则包括出纳员、会计、行长等角色，用户的职责完全由其承担的角色来决定，当其承担的角色发生改变，其职责也会随之改变。信息系统是现实世界的反映，因而在信息系统中一个用户所能访问资源的情况应该随着用户在系统中角色的改变而改变。

基于角色的访问控制(Role-Based Access Control，RBAC)是 20 世纪 90 年代 NIST (National Institute of Standards and Technology)提出的访问控制策略，这种技术能够减少授权管理的复杂性、降低管理开销，而且还能为管理员提供一个比较好的实现复杂安全政策的环境。目前这一访问控制模型已被广为接受。

RBAC 中的基本元素包括用户、角色和权限，RBAC 的核心思想是将访问权限分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的权限。所谓角色(Role)是一个或一群用户在组织内可执行的操作的集合。用户通过角色与相应的访问权限相联系，用户权限是其所拥有角色权限的并集，脱离了角色用户将不存在任何访问权限。角色相当于工作部门中的岗位、职位或分工。一个角色可以有多个权限(对多个资源的访问权)；一个角色可以对应多个用户(相当于一个岗位可以有多个职员)。

【例 5-4】　在学院教务系统中，假设用户有学生 Stud1，Stud2，Stud3，…，Stud*j*，有教师 Tch1，Tch2，Tch3…Tch*i*，有教务管理人员 Mng1，Mng2，Mng3，…，Mng*k*，用户数量众多，在为用户授权时，可以定义如下角色，TchMN＝{查询成绩、上传所教课程的成绩}，Stud MN＝{查询成绩、反映意见}，MngMN＝{查询、修改成绩、打印成绩清单}，为用户分配相应的角色，一旦某个用户成为某角色的成员，则此用户可以完成所具有的职能。

角色由系统管理员定义，角色成员的增减也只能由系统管理员来执行，即只有系统管理员有权定义和分配角色。

2. RBAC 模型

由于 RBAC 采用的很多方法在概念上接近于人们社会生活的管理方式，所以相关的研究和应用发展得很快。从 1996 年发展至今，专家们已经提出了一系列 RBAC 模型，这里主要探讨美国 George Mason 大学提出的 RBAC96 模型，该模型为开发实际的应用系统提供了一个总方针，并为 RBAC 用户提供了评判系统的标准，具体包括 RBAC0、RBAC1、RBAC2、RBAC3 四个模型，其中：

RBAC0—基本模型，规定了任何 RBAC 系统所必需的最小需求。

RBAC1—在 RBAC0 的基础上增加了角色等级(Role Hierarchies)的概念。

RBAC2—在 RBAC0 的基础上增加了限制(Constraints)的概念。

RBAC3—包含了 RBAC1 和 RBAC2，依传递性也间接包含了 RBAC0。

美国国家标准和技术研究所(NIST)已经基于RBAC96制定了RBAC标准,它将RBAC主要分为核心RBAC、有角色继承的RBAC和有约束的RBAC三类。

1) 核心RBAC模型

核心RBAC模型包括6个基本集合:用户集USERS、对象集OBJECTS、操作集OPERATORS、权限集PERMISSIONS、角色集ROLES和会话集SESSIONS,如图5.5所示。USERS中的用户可以执行操作,是主体;OBJECTS中的对象是系统中被动的实体,主要包括被保护的信息资源;对象上的操作构成了权限,因此PERMISSIONS中的每个元素涉及来自OBJECTS和OPERATORS的两个元素,ROLES是RBAC的中心,通过它将用户与特权联系起来,SESSIONS包括了系统登录或通信进程和系统之间的会话。以下具体给出将上述集合关联在一起的操作,通过这些操作,用户被赋予了相应的权限或获得了相应的状态。

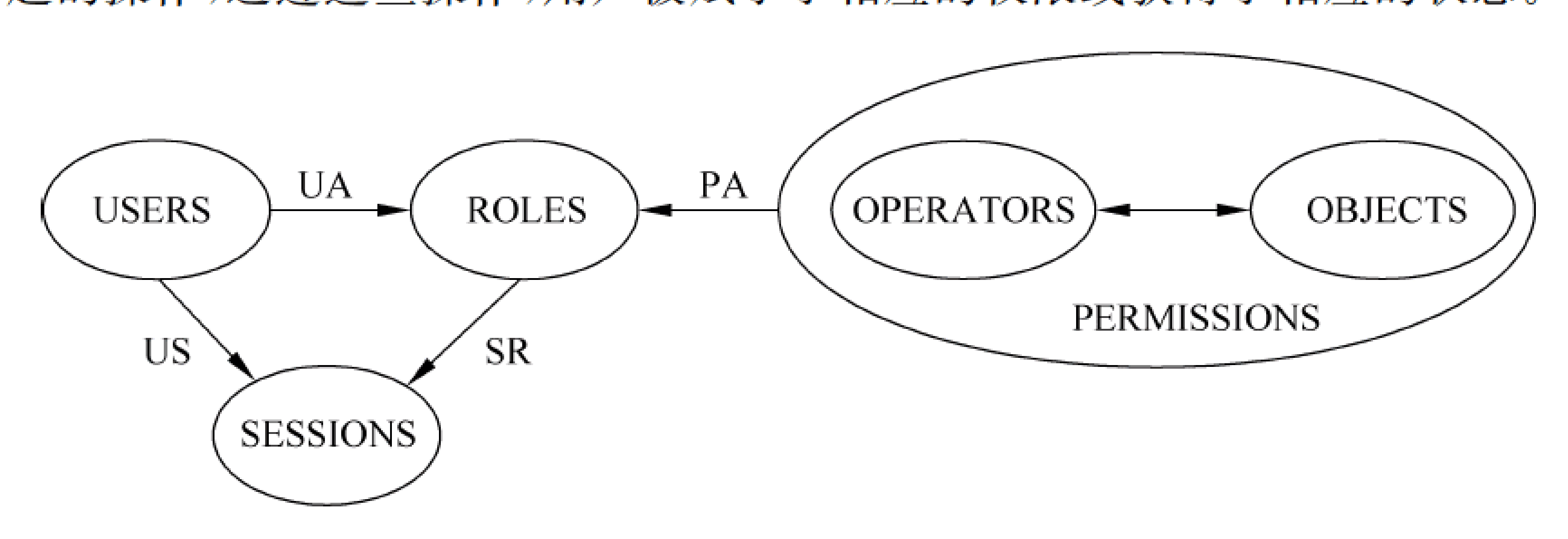


图5.5 核心RBAC中集合及其关系

(1) 用户分配(UA,User Assignment)

UA⊆USERS×ROLES中的元素确定了用户和角色之间多对多的关系,记录了系统为用户分配的角色。若对用户u分配角色r,则UA=UA∪(u,r)。

(2) 特权分配(PA,Permission Assignment)

PA⊆PERMISSION×ROLES中的元素确定了权限和角色之间多对多的关系,记录了系统为角色分配的权限。若把权限p分配给角色r,则PA=PA∪(p,r)。

(3) 用户会话

US⊆USERS×SESSIONS中的元素确定了用户和会话之间的对应关系,由于一个用户可能同时进行多个登录或建立多个通信连接,这个关系是一对多的。

(4) 激活/去活角色

若某个用户属于某个角色,与之对应的会话可以激活该角色,SR⊆SESSIONS×ROLES中的元素确定了会话与角色之间的对应关系,此时该用户拥有与该角色对应的权限。用户会话也可以通过去活操作终止一个处于激活状态的角色。

总之,在RBAC中,系统将权限分配给角色,用户需要通过获得角色来得到权限。

2) 有角色继承的RBAC模型

有角色继承的RBAC模型是建立在以上核心RBAC基础上的,它包含核心RBAC的全部组件,但增加了角色继承(Role Hierarchies,RH)操作,如图5.6所示。如果一个角色r1继承另一个角色r2,r1也有r2的所有权限,并且有角色r1的用户也有角色r2。

RBAC标准包括两种方式的继承:一种是受限继承,一个角色只能继承某一个角色,不支持继承多个角色;另一种是多重继承,一个角色可以继承多个角色,也可以被多个角色继承。这样,角色的权限集不仅包括系统管理员授予该角色的权限,还有其通过角色继承获得

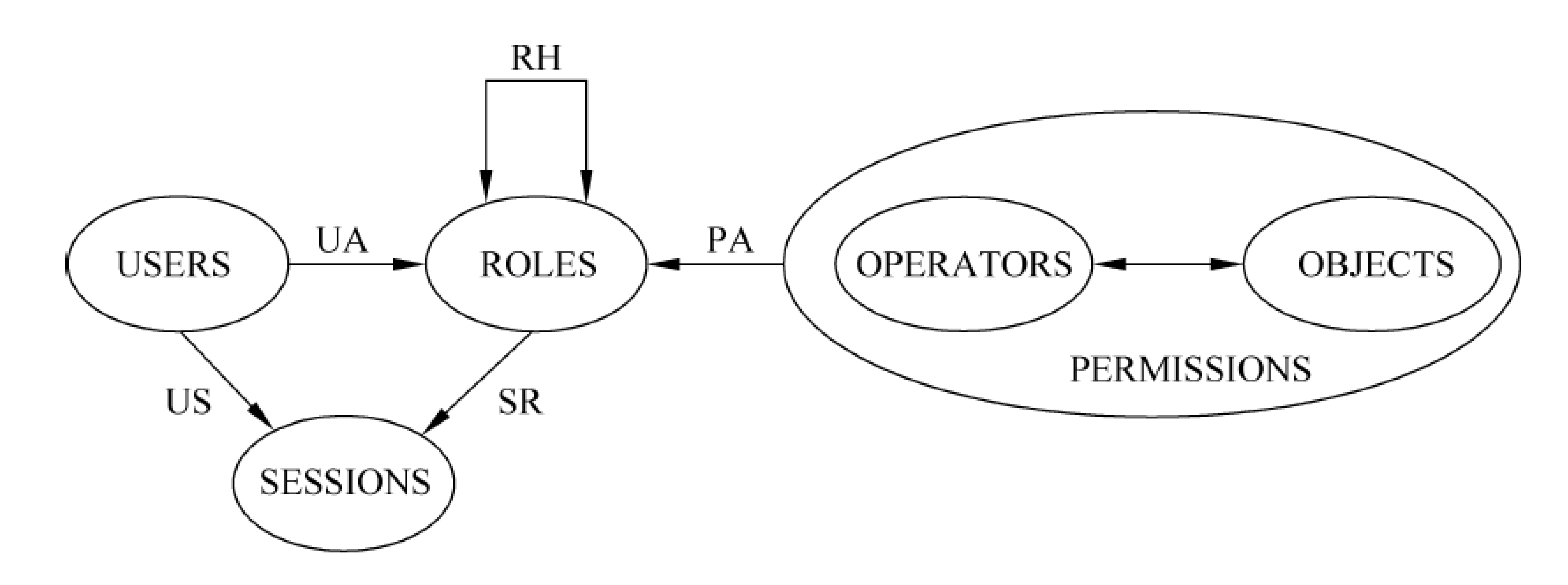


图 5.6　有角色继承的 RBAC 中集合及其关系

的权限，而对应一个角色的用户集不仅包括系统管理员分配的用户，还包括所有直接或间接继承该角色的其他角色分配的用户。

3）有约束的 RBAC 模型

有约束的 RBAC 模型通过提供职责分离机制进一步扩展了以上有角色继承的 RBAC 模型，如图 5.7 所示。职责分离是有约束的 RBAC 模型引入的一种权限控制方法，其目的是为了防止用户超越其正常的职责范围，例如在银行业务中，授权付款与实施付款应该是分开的职能操作，否则可能发生欺骗行为，职责分离主要包括静态职责分离和动态职责分离。

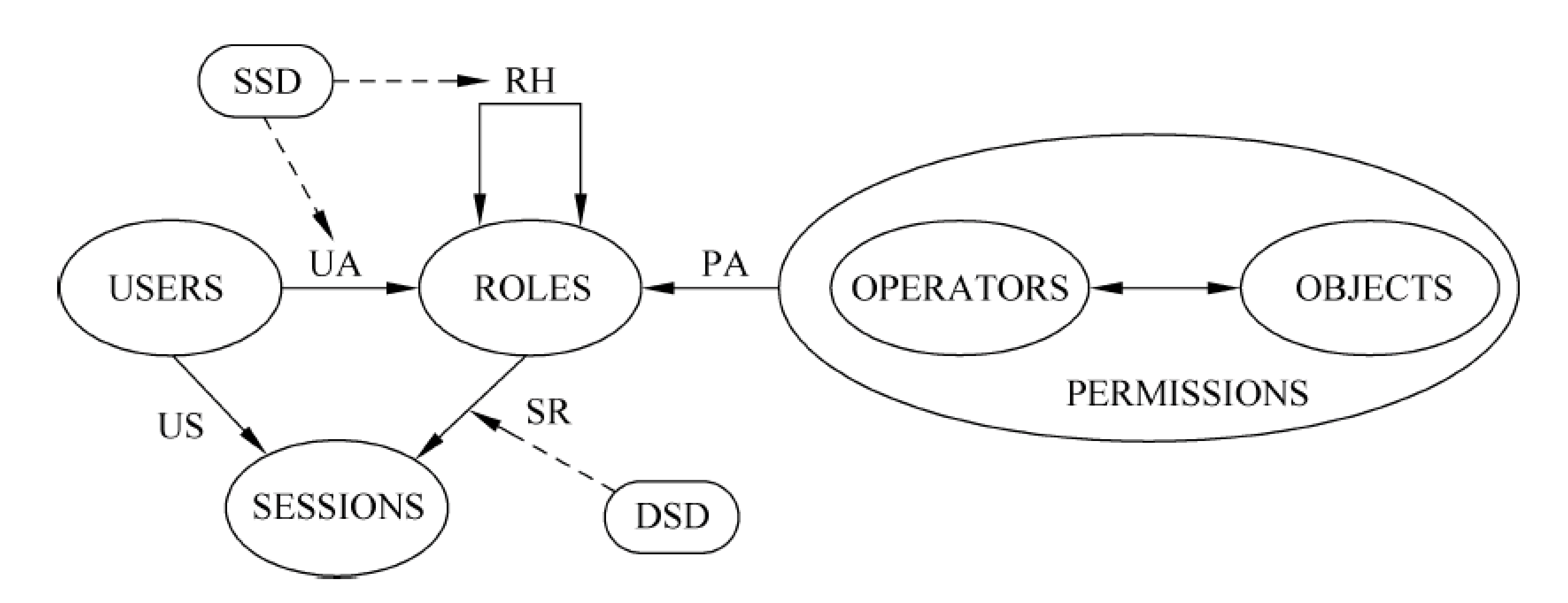


图 5.7　有约束的 RBAC 中集合及其关系

（1）静态职责分离

静态职责分离(Statistic Separation of Duty，SSD)对用户分配和角色继承引入了约束。如果两个角色之间存在 SSD 约束，那么当一个用户分配了其中一个角色后，将不能再获得另一个角色，即存在排他性。由于一个角色被继承将使它拥有继承它的其他角色的全部用户，如果在 SSD 之间的角色存在继承关系，将会违反前述的排他性原则，因此，不能在已经有 SSD 约束关系的两个角色之间定义继承关系。

（2）动态职责分离

动态职责分离(Dynamic Separation of Duty，DSD)引入的权限约束作用于用户会话激活角色的阶段，如果两个角色之间存在 DSD 约束关系，系统可以将这两个角色都分配给一个用户，但是，该用户不能在一个会话中同时激活它们。

3. RBAC 的特点和应用优势

RBAC 具有以下几大特点：

（1）便于授权管理。RBAC 将权限与角色关联起来，用户的授权是通过赋予相应的角

色来完成的。当用户的职责变化时只需要改变角色即可改变其权限；当组织的功能变化或演进时，则只需删除角色的旧功能、增加新功能，或定义新角色，而不必更新每一个用户的权限设置。这极大地简化了授权管理，降低了授权管理的复杂度。

(2) 便于实施职责分离。通过定义角色约束，可以防止用户超越其正常的职责范围，有效地实现职责分离。

(3) 便于实施最小权限原则。最小特权是指用户所拥有的权力不能超过他执行工作所需的权限。实现最小特权原则，需要分清用户的工作职责，确定完成该工作的最小权限集，然后把用户限制在这个权限集范围之内。一定的角色就确定了其工作职责，而角色所能完成的操作蕴含了其完成工作所需的最小权限。用户要访问信息首先必须具有相应的角色，用户无法绕过角色直接访问信息。

正是由于 RBAC 具有灵活性、方便性和安全性的特点，目前在大型数据库管理系统的权限管理中得到了普遍应用，但是，在大型分布式网络环境下，通常无法确知网络实体的身份真实性和授权信息，而 RBAC 无法实现对未知用户的访问控制和委托授权机制，从而限制了 RBAC 在网络环境中的应用。

虽然 RBAC 已在某些系统中得到了应用，但 RBAC 仍处于发展阶段，RBAC 的应用仍是一个相当复杂的问题。

5.3 小　　结

访问控制在身份认证的基础上，根据身份对提出资源访问的请求加以控制。它是对信息系统资源进行保护的重要措施，也是计算机系统最重要和最基础的安全机制。访问控制机制主要包括自主访问控制机制、强制访问控制机制和基于角色的访问控制机制。自主访问控制是计算机系统中实现最多的访问控制机制，其主要特征表现在：主体可以自主地把自己所拥有客体的访问控制权限授予其他主体或从其他主体收回所授予的权限；而 MAC 则根据客体的敏感级和主体的许可级来限制主体对客体的访问，多用于多级军用系统。基于角色的访问控制的基本思想是将访问权限分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的权限，在网络规模变大、用户增多、需求更复杂的情况下，传统的访问控制机制已经不能满足许多企业或组织的安全需求，基于角色的访问控制 RBAC 便明显地显示出其优越性。

习　　题

一、填空题

1. 访问控制的三要素包括________、________、________。

2. 文件的拥有者可以决定其他用户对于相应的文件有怎样的访问权限，这种访问控制是________。

3. 信息系统实现访问控制有多种方式，其中以用户为中心建立起的描述访问权限的表

格，这种方式指的是________。

4．Bell-LaPadula 模型的出发点是维护系统的________，而 Biba 模型与 Bell-LaPadula 模型完全对立，它修正了 Bell-LaPadula 模型所忽略的信息的________问题。

5．访问控制中，访问的发起者称为________，接受访问的被动实体称为________。

6．引用监控器模型中涉及到的基本安全机制有________、________、________。

7．自主访问控制的实现方式有________、________、________。

8．强制访问控制模型主要有________、________。

9．基于角色的访问控制中的基本元素包括________、________、________。

二、选择题

1．下列对访问控制影响不大的是(　　)。

A．主体身份　　B．客体身份
C．访问类型　　D．主体与客体的类型

2．访问控制是指确定(　　)以及实施访问权限的过程。

A．用户权限　　B．可给予哪些主体访问权利
C．可被用户访问的资源　　D．系统是否遭受入侵

3．信息系统实现访问控制有多种方式，其中以用户为中心建立起的描述访问权限的表格，这种方式指的是(　　)。

A．访问控制矩阵　　B．访问控制表
C．访问控制能力表　　D．授权关系表

4．文件的拥有者可以决定其他用户对于相应的文件有怎样的访问权限，这种访问控制是(　　)。

A．自主访问控制　　B．强制访问控制
C．主体访问控制　　D．基于角色的访问控制策略

三、简答题

1．什么是自主访问控制？自主访问控制的实现方法有哪些？

2．什么是强制访问控制？如何利用强制访问控制抵御特洛伊木马的攻击？

3．什么是基于角色的访问控制技术？它与传统的访问控制技术有何不同？

4．简述访问控制的基本概念。

5．有哪几种访问控制策略？

6．访问控制策略制定可以遵循哪些原则？

7．自主访问控制和强制访问控制可以在一个系统中共存吗？

8．与传统的访问控制相比，基于角色的访问控制有哪些优点？

第 6 章　操作系统安全

操作系统是管理计算机硬软件资源、控制程序执行、改善人机界面、提供各种服务、合理组织计算机工作流程、为用户提供良好运行环境的最基本的系统软件。操作系统在信息系统中占有特殊的重要地位,所有其他软件都是建立在操作系统基础之上的。因此操作系统的安全性在计算机信息系统的整体安全性中具有至关重要的作用,为整个计算机信息系统提供底层(系统级)的安全保障。

6.1　操作系统的安全问题

6.1.1　操作系统安全的重要性

计算机信息系统由硬件和软件两部分组成,硬件主要包括处理器、寄存器、存储器及各种 I/O 设备,它们按照用户需求接受和存储信息、处理数据并输出运算结果,是软件运行的物质基础。计算机软件系统主要包括操作系统、应用平台软件、应用业务软件。其中,操作系统是对硬件的第一层软件扩充,用于管理各类计算机资源、控制整个系统的运行,它直接和硬件打交道,并为用户提供接口。应用平台软件主要包括编译程序、数据库管理系统和其他实用程序,用于支持上层应用软件的开发和运行。应用软件层解决用户特定的或不同应用所需要的信息处理问题,如财务系统、航空订票系统等。

操作系统是软件系统的核心,是其他各种软件的基础运行平台,若没有操作系统安全机制的支持,就不可能具有真正的安全性。同时在网络环境中,网络的安全性依赖于各主机系统的安全性,而主机系统的安全性又依赖于其操作系统的安全性。因此,从计算机信息系统的组成角度分析,操作系统的安全性在计算机信息系统的整体安全性中具有至关重要的作用,操作系统为整个计算机信息系统提供系统级的安全保障,没有操作系统的安全性,信息系统的安全性是没有基础的。

6.1.2　操作系统面临的安全问题

操作系统是一种系统软件,不可避免存在缺陷,导致遭受各类安全威胁。美国计算机应急响应组(Computer Emergency Response Term,CERT)提供的安全报告表明:计算机信息系统的很多安全问题都是源于操作系统的安全性。威胁操作系统安全的因素很多,主要有以下几种:

(1) 网络攻击。常见的网络攻击形式有缓冲区溢出攻击、拒绝服务攻击、ARP 攻击、计算机病毒、木马等,严重威胁操作系统的安全,例如,恶意代码(如 Rootkit、逻辑炸弹等)可以直接使系统受到感染而崩溃,也可以使应用程序或数据文件受到感染,造成程序和数据文件的丢失或被破坏,严重的会使系统瘫痪或崩溃。

（2）隐蔽信道(Covert Channel)。隐蔽信道可定义为系统中不受安全策略控制的、违反安全策略的信息泄露路径，一般可分为存储隐蔽信道和时间隐蔽信道，它们都是通过共享资源来传递秘密信息的，隐蔽信道会泄露系统信息、破坏系统的机密性。

（3）用户的误操作。例如，用户无意中删除了系统的某个文件，无意中停止了系统的正常处理任务，这样的误操作会影响系统的稳定运行，严重的会使系统崩溃。

一个有效、可靠的操作系统必须提供相应的保护措施，消除或限制网络攻击、隐信道、误操作等对系统构成的安全隐患。

6.1.3　操作系统的安全性设计

操作系统安全涉及两个重要的概念：安全功能(安全机制)和安全保证。不同的操作系统所提供的安全功能可能不同，实现相同安全功能的途径可能也不同。为此，人们制定了安全评测等级，在安全等级评测标准中，安全功能主要说明各安全等级所需实现的安全策略和安全机制的要求，而安全保证则是描述通过何种方法保证操作系统所提供的安全功能达到了确定的功能要求，安全保证可以从系统的设计和实现、自身安全、安全管理等方面进行描述，也可以借助配置管理、发行和使用、开发和指南文档、测试和脆弱性评估等方面所采取的措施来确立产品的安全确信度。

从安全功能和安全保证在安全评价准则中的组织方式来看，美国 TCSEC 标准将安全功能和安全保证(有的文档称为安全保障)合在一起，共同将计算机信息系统安全保护能力从低到高划分为 D、C、B、A 四类；美国国家标准与技术协会和国家安全局联合开发的联邦标准以及欧洲的 ITSEC 标准，则把安全特性与保障能力分离成两个独立的部分；美、加、英、法、德、荷六国的 7 个组织联合开发的公共准则 CC 标准也采用安全功能和安全保证相独立的理念，即把一个计算机安全产品应该具有的安全特性与为确保这些安全特性的正确实现而采取的安全措施作为两个独立的内容进行分别对待。

我国的 GB 17859-1999 的制定主要参考了美国 TCSEC 标准，同 TCSEC 标准一样，也是将安全功能和安全保证合在一起，共同对安全产品进行评价，但在安全保证方面的要求不太明显。GB/T 18336-2001 则主要参考了国际标准 CC，将安全功能和安全保证独立开来，分别要求，因此，GB 17859 主要对安全功能进行了要求，而 GB/T 18336 则把安全保证作为独立的一部分进行要求和评测。

从安全功能角度，操作系统安全的主要目标如下：

（1）标识系统中的用户并进行身份鉴别。

（2）依据系统安全策略对用户的操作进行存取控制，防止用户对计算机资源的非法存取。

（3）监督系统运行的安全。

（4）保证系统自身的安全性和完整性。

实现操作系统安全目标需要建立相应的安全机制，包括存储保护、用户标识与鉴别、访问控制、最小权限管理、可信路径、安全审计等。在后续章节中会详细介绍。

为了理解操作系统提供的安全保护机制，下面首先简要介绍操作系统的基础知识。

6.2 操作系统基础知识

6.2.1 操作系统的形成和发展

1. 人工操作阶段

从计算机诞生到20世纪50年代中期的机器属于第一代计算机，构成计算机的主要元件是电子管，计算机的运算速度慢，只有几千次/秒，操作系统尚未出现。这一时期，完全由人工控制程序的装入和执行，用户首先用机器语言和符号语言编制程序，并将程序和数据打孔在卡片或纸带上，利用卡片或纸带输入机将程序和数据输入到计算机，然后启动程序运行，程序员通过控制台上的按钮、开关和氖灯操纵和控制程序，程序运行完毕取走计算结果。

这种计算方式的缺点是：用户独占全机资源串行计算，造成资源利用率不高、系统效率低下；人工干预环节较多，不但极其浪费时间，而且极易出错。

随着采用晶体管的第二代计算机的出现，计算机的运算速度大大提高，如果仍采用手工操作方式，将会极大降低计算机的运行效率。例如，一个作业在速度为1千次/s的计算机上执行，假设运行时间为1h，而建立作业和全部人工操作时间需要5min，则两者之比为12∶1，若该作业在6万次/s的计算机上执行，运行时间仅需1min，两者之比变为1∶5，这一比值的变化说明了有效机时的惊人浪费，手工操作的低速度已无法与计算机的高速度相匹配，为了解决这一矛盾，唯一的解决方法就是摆脱手工操作，实现作业之间的自动过渡。

2. 管理程序阶段

早期程序执行的每一步都需要人工干预和辅助，导致大量的计算机时间被消耗，使得极其昂贵的硬件设备只能低效使用。由于每个程序的执行通常包括装入、汇编/编译、执行、输出等几个步骤，为了减少作业执行的人工干预，出现了作业控制语言（Job Control Language，JCL），利用作业控制语言描述对程序执行的控制步骤，将作业控制、程序、数据构成作业提交给计算机操作员，待收集一批作业后通过输入设备（纸带输入机或读卡机）存到磁带上，借助于一个管理程序实现作业装入、执行以及作业之间的自动切换，减少人工操作和干预，让计算机尽可能连续工作。管理程序可以认为是初级操作系统。这个时期由于系统对作业的处理都是分批进行的，且内存中始终只保留一个作业，故称为单道批处理。

3. 多道程序设计和操作系统的形成

在早期的单道批处理系统中，主存中仅有单个作业在运行，CPU和其他硬件串行工作，致使系统中仍有许多资源空闲，设备利用率低。如果能在内存中保留多个程序，当某一程序等待外围设备传输时，让另一道程序运行，就能使CPU得到充分利用，这就是多道程序设计的思想。在20世纪60年代，有两项技术取得了突破性进展：中断和通道。所谓通道，实质是一台功能单一、结构简单的I/O处理机，它独立于CPU，并直接控制外部设备与主机之间的信息传送。当需要进行I/O操作时，CPU启动通道，之后CPU继续执行程序，通道就可以独立工作，通道执行通道程序，完成外部设备的启动、关闭以及信息传输等操作，在传送规定的数据后，它向CPU发出中断请求，CPU终止正在运行的程序，进行中断处理，处理完毕后，返回中断点继续工作。通道和中断技术结合起来，实现了CPU和其他硬件设备的并

行工作，为实现多道程序设计的实现提供了基础。

多道程序设计(Multiprogramming)是指允许多个作业同时进入计算机系统的主存并启动交替计算的方法。也就是说，主存中多个相互独立的程序同时处于开始和结束之间，从宏观上看是并行的，多道程序都处于运行过程中，从微观上看是串行的，各道程序轮流占用CPU 以交替执行。引入多道程序设计技术，可以提高 CPU 的利用率，充分发挥计算机硬件的并发性，现代计算机系统都采用多道程序设计技术。将多道程序技术引入批处理系统，就成为多道批处理系统。

虽然多道批处理系统大大提高了资源利用率，但用户使用计算机不方便，无法干预程序的运行，不利于程序的调试和排错。20 世纪 60 年代末、70 年代初出现了分时操作系统，并得到迅速推广，所谓分时系统就是在主机上连接多台终端，用户通过自己的终端以问答的方式干预程序运行，系统将处理时间划分为小的时间间隔，轮流分给联机终端上的作业，每个终端上的用户请求都能得到快速响应，好像用户独自占用计算机系统一样。批处理系统、分时操作系统的出现，标志着操作系统的正式形成。

6.2.2　操作系统的分类

根据功能、特点和使用方式的不同，可以将操作系统分为以下类型：

(1) 批处理操作系统；

(2) 分时操作系统；

(3) 实时操作系统；

(4) 微机操作系统。

1. 批处理操作系统

批处理操作系统服务于被称为批(Batch)的作业，作业是把程序、数据连同作业说明书组织起来的任务单位，把批中的作业预先输入作业队列，由操作系统按照作业说明书的要求调度和控制作业的执行，大幅度减少人工干预，形成自动转接和连续处理的作业流，由操作员在计算结束后把运算结果返回用户。用户在提交作业直至计算结果之前，不再和计算机及其作业交互。

早期批处理的作业控制说明使用作业控制语言 JCL 书写，在现代操作系统中，执行批作业的控制流采用文件形式表示，如 UNIX 中的 Shell、Windows 中的 autoexec.bat 文件，用户在这类文件中定义一系列操作系统命令，在无用户干预的情况下自动、连续地执行控制文件，在批作业完成后，结果被打印输出并返回给用户。

批处理操作系统的缺点是作业的周转周期延长，不具备交互计算的能力，不利于程序的调试和排错。

2. 分时操作系统

允许多个联机用户同时使用一个计算机系统进行交互式计算的操作系统称为分时操作系统。其实现思路如下：用户在各自的终端上进行会话，程序、数据和命令均在会话过程中提供，以问答的方式控制程序的执行。系统把处理器的时间划分成时间片，轮流分配给各个联机终端，若时间片完成则产生时钟中断，控制权转至操作系统并重新进行调度。由于调试程序的用户常常只发送简短的命令，这样的请求总能得到快速响应，好像用户独占计算机系

统一样，用户直接控制程序的运行，便于程序调试和排错。

3. 实时操作系统

虽然多道批处理操作系统和分时操作系统能够获得较佳的资源利用率和快速响应时间，使计算机的应用范围日益扩大，但它们难以满足实时控制和实时信息处理的需要，于是便产生了实时操作系统。实时操作系统是指当外部事件或数据产生时，能够对其予以接收并以足够快的速度进行处理，所得结果能够在规定时间内控制生产过程或对控制对象做出快速响应，并控制所有实时任务协调运行的操作系统。因而，提供及时的响应和高可靠性是其主要特点。

在实时操作系统中要有实时时钟管理，以便对实时任务进行实时处理。系统中通常存在若干实时任务，往往通过"队列驱动"或"事件驱动"开始工作，当系统接收某个外部事件之后，驱动实时任务以完成相应的处理和控制。

4. 微机操作系统

微型计算机的出现引发计算机产业革命，使其进入社会生活的各个领域，拥有巨大的使用量和最广泛的用户群。从 20 世纪 70 年代中期到 80 年代早期，微型计算机上运行单用户单任务操作系统，如 CP/M、CDOS(Cromenco 磁盘操作系统)、MDOS(Motorola 磁盘操作系统)和早期的 MS-DOS(Microsoft 操作系统)。从 20 世纪 80 年代中期到 90 年代初，微机操作系统开始支持单用户多任务和分时操作，其中以 MP/M、XENIX 和 Windows 9x 系列为典型代表。

近年来，微机操作系统得到进一步发展，以 UNIX、Windows、OS/2 和 Linux 为代表的多用户多任务微机操作系统具有 GUI、虚拟存储管理、网络通信支持、数据库支持、多媒体支持等功能，具有开放性、通用行、高性能等特点。

5. 网络操作系统

计算机网络通过通信设施将地理上分散且具有自治功能的多个计算机系统互联起来，网络操作系统能够控制计算机在网络中传送信息和共享资源，并为网络用户提供所需的各种服务，其主要功能有网络通信、资源管理、网络管理和网络服务等。目前，主要的三种网络操作系统是 UNIX、NetWare 和 Windows NT。UNIX 是唯一的跨平台操作系统，Windows NT 工作在微型计算机和工作站上，NetWare 则主要面向微型计算机。

尽管尚无严格定义，但一般认为操作系统是管理系统资源、控制程序执行、改善人机界面、提供各种服务，并合理组织计算机工作流程和为用户方便而有效地使用计算机提供良好运行环境的最基本的系统软件。

6.2.3 操作系统功能

从资源管理的观点来看，操作系统的主要功能包括处理器管理、存储管理、设备管理、文件管理、网络与资源管理、用户接口。

1. 处理器管理

处理器是计算机系统中最为宝贵的资源，应该最大限度地提高其利用率，常常采用多道程序设计技术，组织多个作业同时执行，解决处理器的调度、分配和回收问题。为了做好处理器管理，描述多道程序的并发执行，操作系统引入进程的概念，处理器的分配、调度和执行

都以进程为基本单位。随着并行处理技术的发展,并发执行单位的粒度变细,并发执行的代价降低,在进程的基础上又引入了线程概念。对处理器的管理和调度最终归结为对进程和线程的管理和调度。

2. 存储管理

存储管理的主要任务是管理主存资源,为多道程序运行提供有力的支撑,提高存储空间的利用率。具体功能包括以下几个。

(1) 主存分配:根据应用程序的需要向其分配主存资源,在程序运行结束时,回收主存资源。

(2) 地址转换与存储保护:负责把用逻辑地址编程的应用程序装入主存,并将逻辑地址转换成物理地址,同时保证各应用程序互不干扰,不允许它们访问操作系统存储区。

(3) 主存共享:能够让主存中的多个应用程序实现存储共享,提高存储资源的利用率。

(4) 存储扩充:由于受到处理器寻址能力的限制,一台计算机的物理主存容量总是有限的,难以满足大型应用程序的需求,而辅助存储器容量大而且价格便宜,故存储管理应能够从逻辑上扩充主存,把主存和辅存混合起来使用,为用户提供比实际主存容量大得多的虚拟存储器。

3. 设备管理

设备管理的主要任务是管理各种外部设备,完成用户所提出的 I/O 请求;加快数据传输速度,发挥设备的并行性,提高设备的利用率;提供设备驱动程序和中断处理程序,为用户隐蔽硬件操作细节,提供简单的设备使用方法。

4. 文件管理

上述三种管理是针对计算机硬件资源的管理,文件管理则是针对信息资源的管理。通常程序和数据以文件的形式存储在辅助存储器上,操作系统中配置了文件系统,对用户文件和系统文件进行有效的管理,实现按名存取,实现文件的共享、保护,并向用户提供一整套文件操作接口。

5. 网络与资源管理

计算机网络源于计算机技术与网络技术的结合,近些年来,网络的应用范围已经十分广泛。操作系统至少应具有网络资源管理、数据通信管理和网络管理的功能。

6. 用户接口

为了使用户能够灵活、方便地使用计算机硬件和系统所提供的服务,操作系统向用户提供一组使用其功能的手段,称为用户接口,包括程序接口和操作接口,用户通过这些接口能够方便地调用操作系统的功能,有效组织作业及其处理流程,使得整个计算机系统高效运行。

6.2.4　程序接口和系统调用

操作系统通过程序接口和操作接口将其服务和功能提供给用户,如图 6.1 所示。程序接口是操作系统对外提供服务和功能的手段,它由一组系统调用组成,在应用程序中使用“系统调用”可获得操作系统的底层服务,访问或使用系统管理的各种软硬件资源;操作接

口由一组控制命令和作业控制语言组成,控制命令可以在字符型用户界面下通过命令行或批命令的方式实现,这种方式需要牢记各种命令的动词和参数,严格按规定的格式输入命令,既费时又不方便,目前最为流行的联机作业控制方式采用图形用户界面 GUI,通过图标将系统的各项控制功能直观、逼真地表示出来,用户通过选择窗口、菜单、对话框等完成控制和操作。

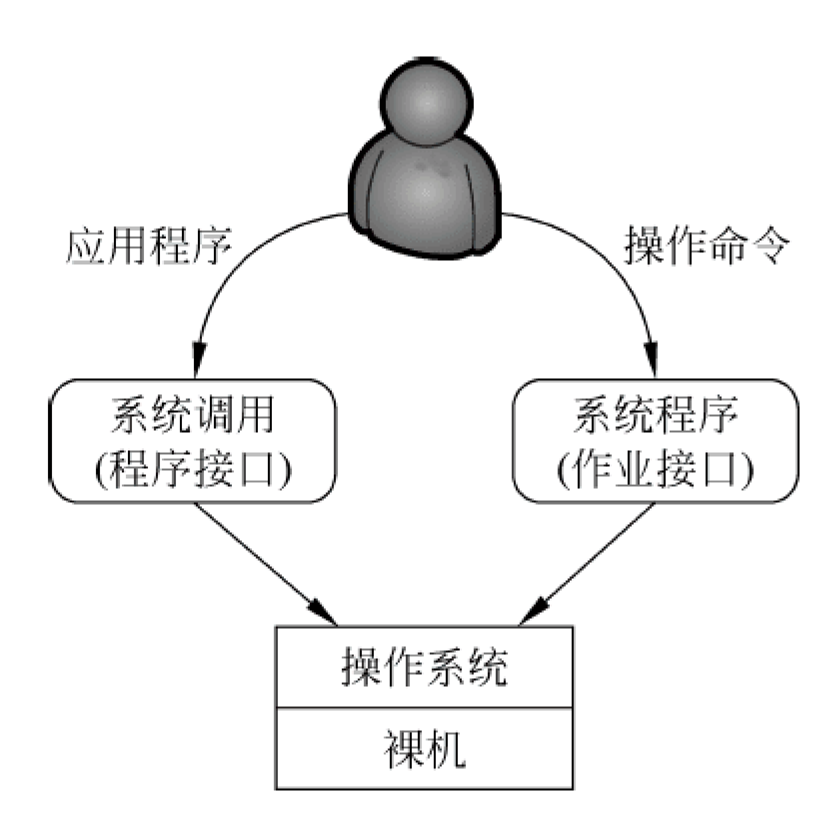


图 6.1 用户和操作系统之间的接口

为了给应用程序的运行创建良好的环境,操作系统内核提供了一系列具备特定功能的内核函数,通过一组称为系统调用(System Call)的接口呈现给用户。系统调用把应用程序的请求传送至内核,调用相应的内核函数完成所需的处理,将处理结果返回给应用程序,如果没有系统调用和内核函数,用户将不可能编写功能强大的应用程序。可以这样认为,内核的主体是系统调用的集合。

操作系统服务之所以通过系统调用的方式供用户使用,其根本原因是为了对系统进行"保护"。程序的运行空间分为内核空间和用户空间,其程序各自按不同的特权运行,在逻辑上相互隔离。应用程序不能直接访问内核数据,也无法直接调用内核函数,它们只能在用户空间操纵用户数据,调用用户空间函数,但是在很多情况下,应用程序需要获得系统服务,这时就必须利用系统提供给用户的特殊接口——系统调用。

当 CPU 执行程序中预设的由访管指令实现的系统调用时,会产生异常信号,通过中断机制,处理器的状态由用户态转变为核心态,进入操作系统并执行相应的内核函数,以获得操作系统服务,当系统调用执行完毕时,控制权返回至发出系统调用的程序,系统调用是应用程序获得操作系统服务的唯一途径。

每种操作系统所提供的一组系统调用虽然功能大同小异,但是其实现细节不尽相同,与具体的机型有关。如果应用程序直接使用系统调用,至少存在两个问题,一是接口复杂、使用困难,二是应用程序的跨平台可移植性受到很大限制,为此 IEEE 开发了 POSIX(Portable Operating System Interface for Computer Environments,计算机环境可移植操作系统接口)标准,其中 POSIX 专门规定内核的系统调用接口标准,操作系统的实现若遵循此标准,那么应用程序在不同操作系统之间就具有可移植性。UNIX/Linux 遵循 POSIX 标准,所以它们是 POSIX 兼容的操作系统。

为了能够在 C 语言程序中使用系统调用,UNIX/Linux 在标准 C 函数库中为每个系统调用构造一个同名的封装函数,屏蔽其下各层的复杂性,这样的封装函数是库函数,它是应

用程序可以直接使用的 API(Application Program Interface,应用程序接口),API 是一个函数的定义,一个 API 的实现可能会用到一个或多个系统调用,也可能若干 API 封装相同的系统调用,也有可能完全不使用系统调用。例如 open()库函数与 open()系统调用一一对应,但 strcpy()库函数没有使用任何系统调用。对于复杂的库函数,通过系统调用转向内核函数只是其工作的一小部分,fprintf()就是这样,它要提供数据缓存和格式编排,最后才通过 write()系统调用把数据写到设备上。UNIX/Linux 操作系统既支持库函数,又公开系统调用,所以 UNIX 应用程序既可以通过_syscall*n*()(*n* 为传递的参数个数)直接使用系统调用,又可通过库函数间接使用系统调用,以此获得操作系统的服务。如图 6.2 所示为应用程序执行 API fprintf()时,调用 C 函数库中的 fprintf()库函数,再调用 C 函数库中的 write()库函数,最终调用内核函数 sys_write()的关系链。

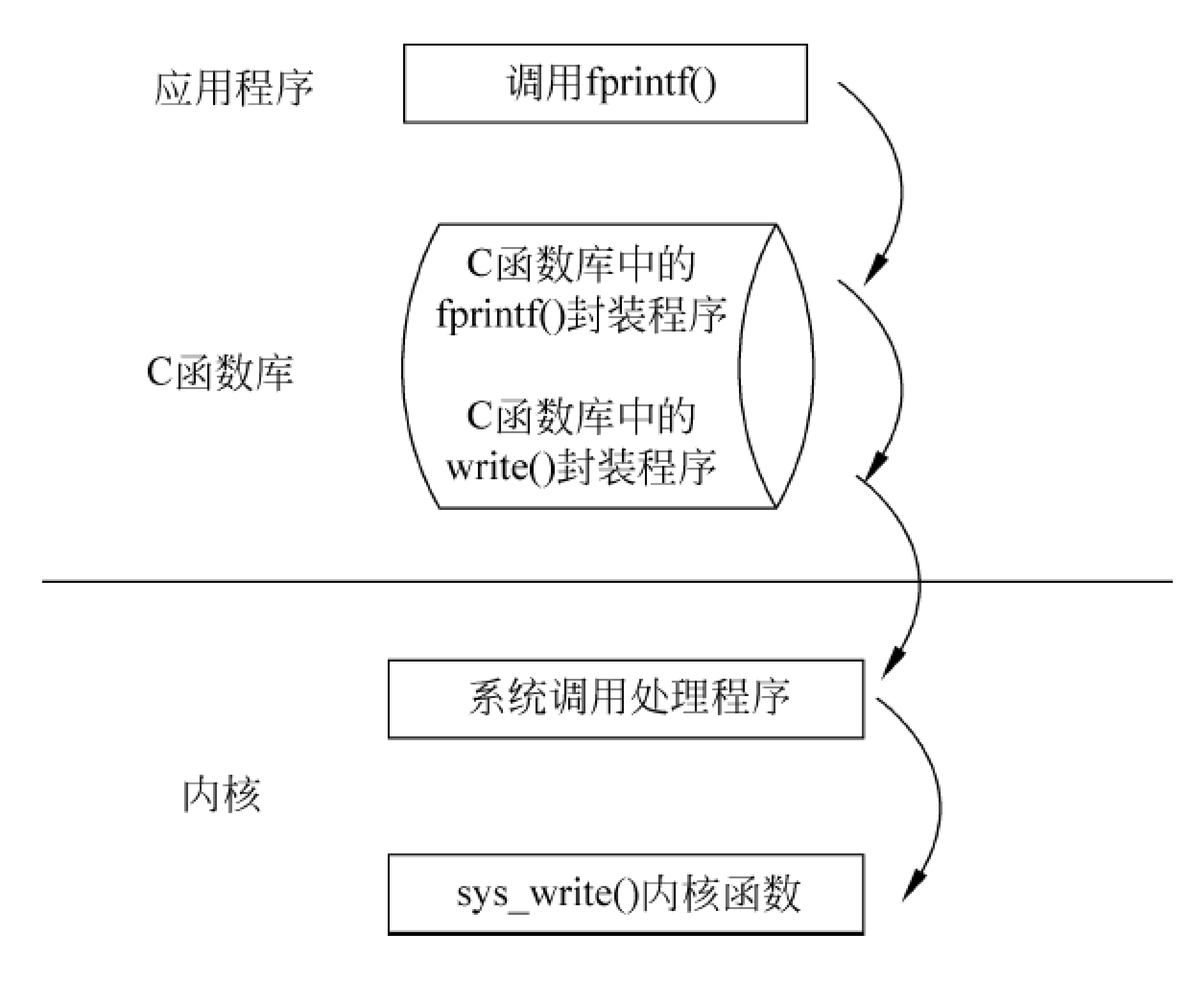


图 6.2　应用程序库函数执行过程

从应用程序的角度来看,库函数与系统调用之间的差别并不重要,但从实现的角度来看,两者之间存在重大区别。使用库函数的好处是可以隐藏访管指令的细节,使得系统调用更像函数调用,对用户屏蔽系统调用,这样在改动内核时不会影响应用程序的正确性。但是库函数属于应用程序,在用户态运行,系统调用属于系统程序,在核心态运行,如果需要的话,用户可以替换库函数,通常却不能替换系统调用。

6.2.5　进程

进程是操作系统中最基本、最重要的概念,进程的概念最早是 1960 年在美国麻省理工学院 MULTICS 和 IBM 公司 CTSS/360 系统中提出和实现的。一般认为,进程是可并发执行的程序在某个数据集合上的一次计算活动,进程是操作系统进行资源分配和保护的基本单位。

从理论的角度看,进程是对当前运行程序活动规律的抽象,从实现的角度看,进程是一种数据结构,用来准确刻画系统动态变化的内在规律,有效地管理和调度在计算机系统主存运行的程序。首先讨论操作系统中引入进程的目的。

1. 引入进程的原因

处理器(CPU)是计算机系统的宝贵资源,需要进行有效管理,早期的单道批处理系统中,主存中仅有单个作业在运行,CPU 和其他硬件设备串行工作,致使系统中仍有许多资源空闲,设备利用率很低。在 20 世纪 60 年代,有两项技术取得突破性进展:中断和通道,使得计算机体系结构由原先的以 CPU 为中心转变成以主存为中心,通道能够产生 I/O 中断,具有中断主机工作的能力,这两种技术相结合,为实现 CPU 和其他硬件设备的并行工作提供了一定的基础,产生了多道程序设计。

多道程序设计是指允许多个作业(程序)同时进入计算机系统的主存并启动交替计算的方法。也就是说主存中多个相互独立的程序均处于开始和结束之间,从宏观上看是并行的,多道程序都处于运行过程中,从微观上看是串行的,各道程序轮流占用 CPU 以交替执行。引入多道程序设计,可以提高 CPU 的利用率,充分发挥计算机硬件的并行性,现代计算机系统都采用多道程序设计技术。

在现代计算机系统中,I/O 操作较慢而 CPU 运行速度快,故程序运行时花费在 I/O 操作上的时间最多,而程序在执行 I/O 时并不需要 CPU,从第二代计算机开始,系统具有 CPU 和设备并行工作的能力,采用多道程序设计技术可以提高系统资源利用率。

【例 6-1】 求解某个数据问题,要求从输入机(运转速度为 6400 个字符/s)输入 500 个字符,经处理(费时 52ms)之后,将结果(假定为 2000 个字符)存储到磁带上(磁带机的运转速度为 105 个字符/s),然后,再读取 500 个字符进行处理,直至所有数据处理完毕为止。如果 CPU 不具备设备并行工作的能力,那么 CPU 的利用率为:

$$52/(78+52+20)\approx 35\%$$

可以看出系统效率之所以不高是因为当输入机输入 500 个字符之后,处理器只用 52ms 进行处理,而第二批输入数据尚需等待 98ms 才能输入完毕,在此期间 CPU 一直空闲。这说明在单道程序工作时,计算机系统各部件的利用率并未得到充分的发挥。为了提高系统效率,让计算机同时接受两道计算题,当第一道程序等待设备进行数据传输时,让第二道程序运行,以缩短 CPU 空闲等待时间,那么 CPU 的利用率将得以提高。例如,计算机在接收上述例题时还接收另一道计算题,从另一台磁带机上输入 2000 个字符,经 42ms 的处理之后,从行式打印机(运转速度为 1350 行/min)上输出两行。

当两道程序同时进入主存时,计算过程如图 6.3 所示,其中 P_1 表示程序甲,该程序首先占用 CPU 对输入的 500 个字符进行处理,下次处理要等待 98ms,因而这段时间内 CPU 是空闲的。系统调度程序乙(用 P_2 表示)工作,它从磁带机上输入 2000 个字符,然后对这批数据进行处理,相应的设备和 CPU 操作都是并行的,不难算出,此时 CPU 的利用率为:

$$(52+42)/150\approx 63\%$$

所以,多个程序同时进入主存执行计算比程序逐个串行计算的 CPU 利用率要高,因为当某个程序因故不能继续运行时,系统会把 CPU 分配给另一个程序,使得 CPU 和设备尽量处于忙碌状态,这就是采用多道程序设计的主要原因。

在多道程序环境下,处理器在各程序之间来回切换,程序是并发执行的,程序的任意两条指令之间都可能发生事件而引发程序切换,因而程序的执行可能不是连续的而是走走停停。此外,程序的并发执行又会引发资源共享和竞争问题,造成并发执行的各个程序之间可能存在制约关系,执行的程序不再处于封闭环境中,出现许多新特征。而程序自身只是计算

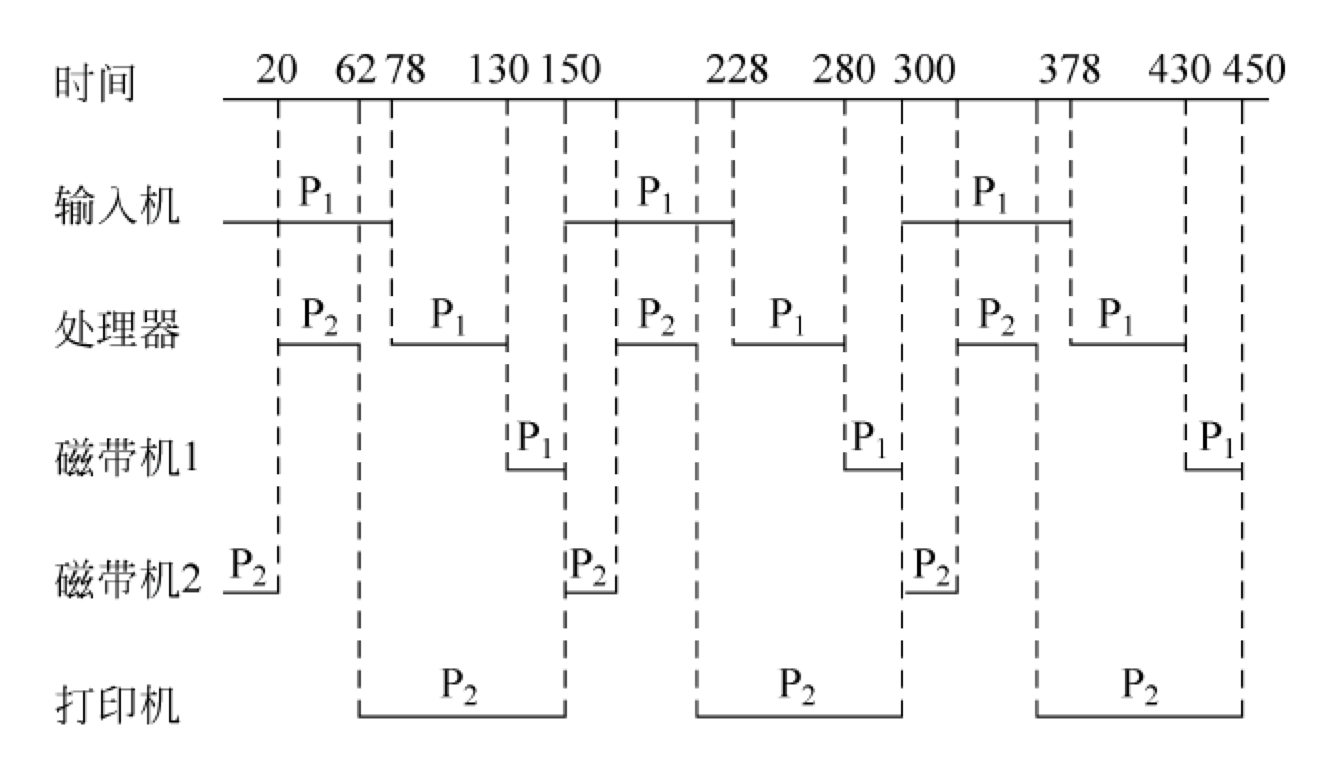


图 6.3　两道程序运行时处理器的利用率

任务的指令和数据的描述，这种静态的概念无法刻画程序的并发性，系统要寻找能够描述程序动态执行过程的概念，这就是进程。进程是并发程序设计的一种有力工具，操作系统中进程概念能较好地刻画系统内部的“动态性”，发挥系统的“并发性”，从而提高资源利用率。

2. 进程的状态和转换

进程从因创建而产生直至撤销而消亡的整个生命周期中，有时占用处理器执行，有时虽然运行但分不到处理器，有时虽然处理器空闲但因等待某个事件发生而无法执行，这一切说明了进程和程序的不同，进程是活动的且有状态变化，状态及状态之间的转换体现了进程的动态性。为了便于系统管理，一般来说，按照进程在执行过程中的不同情况至少定义以下三种进程状态：

(1) 运行态(Running)：进程占用处理器运行的状态。

(2) 就绪态(Ready)：进程具备运行条件，等待系统分配处理器以便其运行的状态。

(3) 等待态(Wait)：又称阻塞态(Blocked)或睡眠态(Sleep)，是指进程不具备运行条件，正在等待某个事件完成的状态。

处于运行态的进程个数不能大于处理器的个数，处于就绪态和等待态的进程可以有多个。进程通常在创建之后处于就绪态，进程在执行过程中的任一时刻必然处于上述三种状态之一，进程在执行过程中其状态将发生改变，如图 6.4 所示。

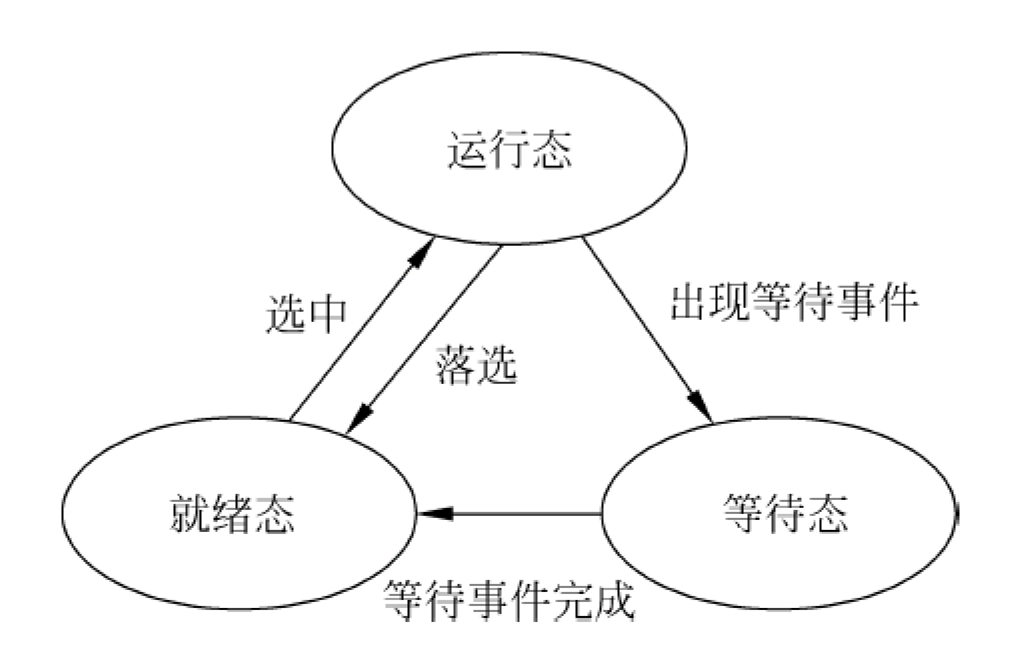


图 6.4　进程基本状态转换

引起进程状态转换的具体原因有以下几点：

(1) 运行态-等待态：运行进程等待使用某种资源或某事件发生，如等待设备传输数据或人工干预。

(2) 等待态-就绪态：所需资源得到满足或某事件已经完成，如设备传输数据结束或人工干预完成。

(3) 运行态-就绪态：运行时间片到时或出现更高优先级的进程时，当前进程被迫让出处理器。

(4) 就绪态-运行态：当CPU空闲时，调度程序选中一个就绪进程执行。

到目前为止，总是假设所有进程都在主存中，事实上，可能出现这样的情况，由于不断地创建进程，系统资源特别是主存资源已经不能满足进程运行的要求，此时必须把某些进程挂起(Suspend)，置于磁盘对换区中，释放其占用的某些资源，暂时不启用低级调度，起到平滑系统负载的目的，也可能系统出现某种故障，需要暂时挂起一些进程，以便在故障消除后，解除挂起并恢复进程的运行，总之引起进程挂起的原因多种多样。

如图6.5所示为具有挂起进程功能的系统中的进程状态，两个新状态是挂起就绪态(Ready Suspend)和挂起等待态(Blocked Suspend)。挂起就绪态表明进程具备运行条件，但目前在辅助存储器中，只有当进程被对换到主存时才能调度执行；挂起等待态则表明进程正在等待某一事件发生且进程在辅助存储器中。

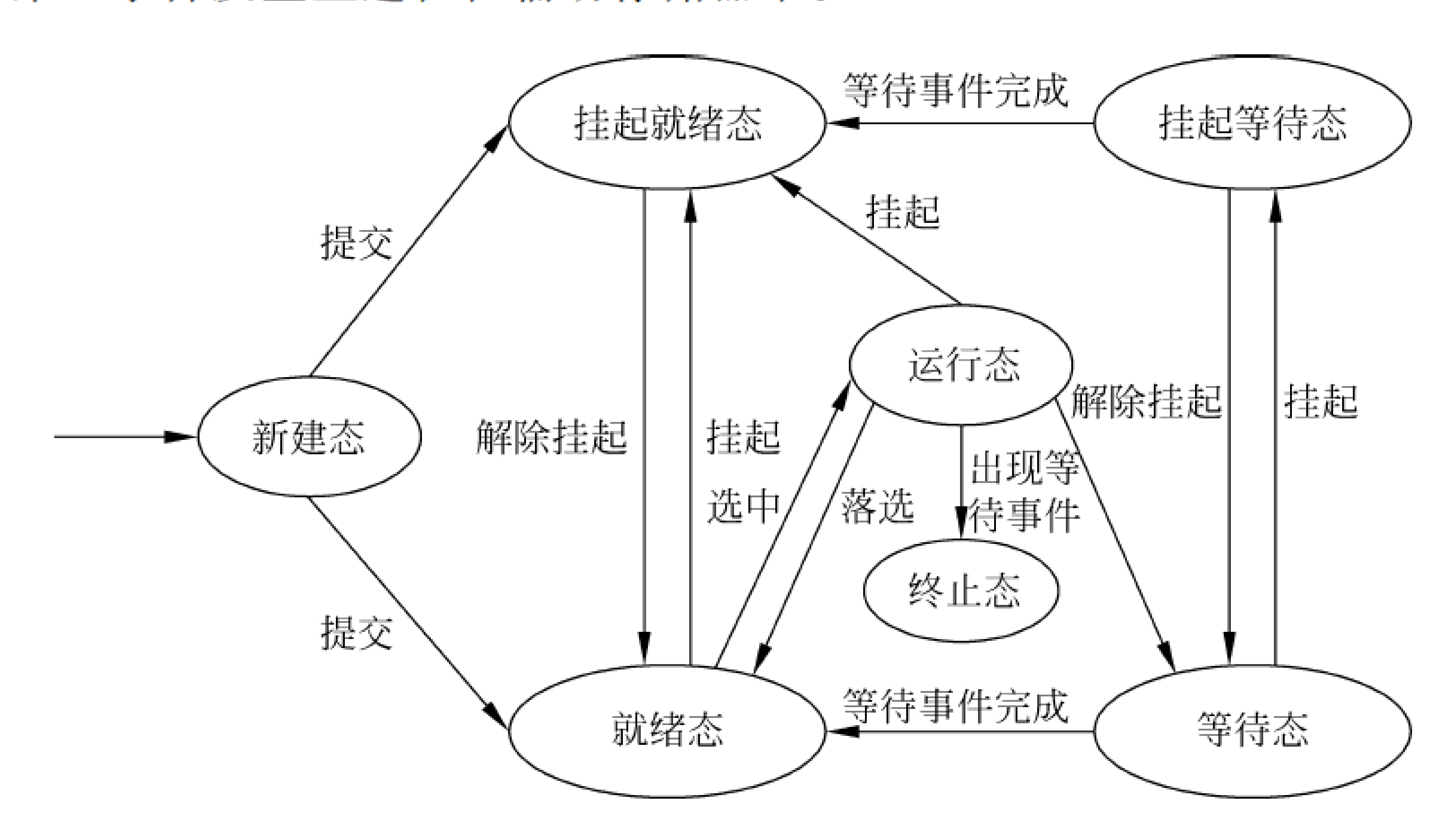


图6.5 具有挂起进程功能的系统中进程状态转换

引起进程状态转换的具体原因如下：

(1) 等待态→挂起等待态：如果当前不存在就绪进程，系统根据资源分配情况和性能要求，选择等待态进程对换出去，使之处于挂起等待态。

(2) 挂起等待态→挂起就绪态：导致进程等待的事件完成后，相应的处于挂起等待态的进程将转换为挂起就绪态。

(3) 挂起就绪态→就绪态：当主存中不存在就绪进程，或者挂起就绪态进程具有比就绪态进程更高的优先级时，系统将把挂起就绪态进程换回主存并转换成就绪态。

(4) 就绪态→挂起就绪态：系统根据当前资源分配情况和性能要求，决定把就绪态进程对换出去，使之处于挂起就绪态。

(5) 挂起等待态→等待态：进程等待事件发生时，原则上无须将其调入主存，但当某些进程撤销后，主存拥有足够的自由空间，而某个挂起等待态进程具有较高的优先级，且系统得知导致它阻塞的事件即将结束，便可能发生这一类状态变化。

(6) 运行态→挂起就绪态：当一个具有较高优先级的挂起等待进程所等待的事件完成

后，它需要强占 CPU，而此时主存空间不够，可能会导致正在运行的进程转换为挂起就绪态。另外，运行态进程也可自我挂起。

(7) 新建态→挂起就绪态：考虑系统当前资源分配状况和性能要求，决定将新建进程对换出去，使之处于挂起就绪态。

不难看出，挂起进程等同于不在主存的进程，因此，挂起进程不会参与低级调度直到它们被对换进主存。挂起进程具有以下特征：此进程不能立即执行；此进程可能会等待某事件发生，所等待的事件独立于挂起条件，事件结束并不能导致进程具备可执行条件，此进程进入挂起状态是由于操作系统、父进程或进程自身阻止其运行；进程挂起状态的结束命令只能通过操作系统或父进程发出。

3. 进程的描述和组成

1) 进程映像

进程的活动包括占用处理器执行程序以及对相关数据进行操作，因而，程序和数据是进程必需的组成部分，两者刻画进程的静态特性，除此之外，还需要数据结构来刻画进程的动态特征，描述进程状态、占用资源状况、调度信息等，通常使用一种称为进程控制块的数据结构。由于进程的状态在不断发生变化，某时刻进程的内容及其状态集合称为进程映像(Process Image)，包括以下一些要素：

(1) 进程控制块：每个进程绑定一个控制块，用来存储进程的标志信息、现场信息和控制信息。进程创建时建立进程控制块，进程撤销时回收进程控制块，它与进程一一对应。

(2) 进程程序块：进程程序块是被执行的程序，规定进程的一次运行所应完成的功能。

(3) 进程核心栈：每个进程绑定一个核心栈，进程在核心态工作时使用，用来保存中断/异常现场保存函数调用的参数和返回地址等。

(4) 进程数据块：进程数据块是进程的私有地址空间，存放各种私有数据，用户栈也要在数据块中开辟，用于在函数调用时存放栈帧、局部变量等参数。

可见，每个进程由 4 个要素组成：控制块、程序块、核心栈和数据块。

2) 进程控制块

每个进程有且仅有一个进程控制块(Process Control Block，PCB)，或称进程描述符，它是进程存在的唯一标识，是操作系统用来记录和刻画进程状态及有关信息的数据结构，是进程动态特征的一种汇集，也是操作系统掌握进程的唯一资料结构和管理进程的主要依据。进程控制块包括进程执行时的情况以及进程让出处理器后所处的状态、断点等信息，一般来说，应包含以下三类信息。

(1) 标识信息

标识信息用于唯一地标识一个进程，分为用户使用的外部标识符和系统使用的内部标志号。系统中的所有进程都被赋予唯一的、内部使用的数值型进程号(通常是 0～32 768 中的正整数)，操作系统内核函数可通过进程号来引用 PCB。常用的标识信息包括进程标识 ID、进程组标识 ID、用户名、用户组名等。

(2) 现场信息

现场信息用于保留进程在运行时存放在处理器现场中的各种信息。进程在让出处理器时，必须将此时的现场信息保存在它的 PCB 中，而当此进程恢复运行时也应恢复处理器现场。现场信息包括通用寄存器内容、控制寄存器内容、栈指针等。

(3) 控制信息

控制信息用于管理和调度进程,包括进程调度的相关信息,如进程状态、等待事件和等待原因、进程优先级、队列指针等;进程组成信息,如正文段指针、数据段指针、进程之间的族系信息,如指向父/子/兄进程的指针;进程间的通信信息,如消息队列指针、所使用的信号量和锁,进程段/页表指针、进程映像在辅助存储器中的地址;CPU 的占用和使用信息,如时间片剩余量、已占用 CPU 时间、进程已执行时间总和、定时器信息、记账信息;进程特权信息,如主存访问权限和处理器特权;资源清单,如进程所需的全部资源、已经分得的资源,例如主存、设备、打开文件表等。

PCB 是操作系统中最重要的数据结构,它包含管理进程所需要的全部信息,PCB 的集合实际上定义了操作系统的当前状态,其使用权和修改权均属于操作系统,包括调度程序、资源分配程序、中断处理程序、性能监视和分析程序等。系统在创建进程时就为它建立 PCB,当进程运行结束被撤销时,将回收其所占用的 PCB。操作系统根据 PCB 对并发执行的进程进行控制和管理,进程借助于 PCB 才能被调度执行。

4. 进程的控制和管理

系统中的进程不断地产生和消亡,进程生命周期的动态变化过程由进程管理程序来控制,对于进程的控制和管理包括创建进程、阻塞进程、唤醒进程、挂起进程、激活进程、终止进程和撤销进程等,这些功能均由系统中的原语实现。原语(Primitive)在核心态执行,是完成系统特定功能的不可分割的过程,它具有原子操作性,其程序段不允许被中断,或者说原语不能并发执行,系统对进程的控制如果不使用原语,就会造成状态的不确定性,不能达到进程控制的目的。下面介绍部分进程控制原语。

1) 进程创建

可以动态地创建新进程,当通过内核为一个程序构造 PCB 并分配地址空间后,就创建了一个进程,进程的创建来源于以下事件:

(1) 提交批处理作业;

(2) 有交互式作业登录终端;

(3) 操作系统创建服务进程;

(4) 已存在的进程创建新进程。

当用户作业被选中进入主存时,需要创建用户进程来完成作业。一个用户进程在请求某种服务时,也需要创建一个或多个子进程或系统进程来为其服务。例如,当用户进程读取卡片上的一段数据时,可能要求创建卡片输入机管理进程。有的操作系统把创建用父子进程的关系来表示,当一个进程创建另一个进程时,生成进程称为父进程,被生成进程称为子进程,父进程可以创建多个子进程,从而形成树状族系关系,一般来说,进程的创建过程如下:

(1) 在进程列表中增加一项,从 PCB 池中申请一个空闲 PCB,为新进程分配唯一的进程标志符。

(2) 为新进程的进程映像分配地址空间,以便容纳进程实体。由进程管理程序确定加载至进程地址空间中的程序。

(3) 为新进程分配除主存空间以外的其他各种资源。

(4) 初始化 PCB,如进程标识符、处理器初始状态、进程优先级等。

(5) 把新进程的状态设置为就绪态,并将其移入就绪进程队列。

(6) 通知操作系统的某些模块,如记账程序、性能监控程序。

不同操作系统创建进程的方式不尽相同,传统 UNIX 系统中是这样处理的:父进程使用 fork()创建子进程,此系统调用不带任何参数,子进程的行为完全由父进程和默认行为所决定,把自己的地址空间制作一个副本,其中包括 user 结构(包含所有打开文件的文件描述符)、正文段、数据段、用户栈和核心栈,即子进程继承父进程的全部资源;同时,系统将子进程的 pid 返回给父进程,向子进程返回 0,这种实现方式使得父子进程易于通信。当然,如果子进程需要,可以使用系统调用 execve()让新程序替换老程序,此后,父子进程之间可以自行其道。

Linux 保留传统的 fork()创建子进程的方法,在调用 fork()创建子进程后,父子进程存在以下关系:调用一次,返回两次,分别返回给父子进程,父子进程是独立的进程,可以并发执行,父子进程具有独立的地址空间,如果父进程改变某个变量的值,子进程将不会看到这个变化,反之亦然。

2) 进程撤销

进程完成特定的工作或出现严重错误之后,操作系统将回收其所占有的地址空间和 PCB,此时就是撤销一个进程。进程撤销的主要原因有:进程运行结束;进程执行非法指令;进程运行时间超过分配的最大时间限额;进程等待时间超过所设定的最长等待时间;越界错误;父进程撤销其子进程;父进程撤销,其所有子进程被撤销;操作系统终止等。

一旦发生上述事件,系统或进程将调用撤销原语来终止进程或子进程,具体步骤如下:

(1) 根据撤销进程的标识号,从相应的队列中查找并移除它;

(2) 将此进程所拥有的资源归还给父进程或操作系统;

(3) 若此进程拥有子进程,先撤销其所有子进程,以防止它们脱离控制;

(4) 回收 PCB,并将其归还至 PCB 池。

3) 进程的阻塞与唤醒

进程阻塞是指进程让出处理器,转而等待一个事件,如等待资源、等待 I/O 操作完成、等待某事件发生等。进程通常自调用阻塞原语来阻塞自己,所以,阻塞是进程的自主行为,是一个同步事件。当等待事件完成时,会产生一个中断激活操作系统,在系统的控制之下将被阻塞的进程唤醒。等待事件完成是指 I/O 操作结束、某个资源可用或所期待的事件发生。进程的阻塞和唤醒显然是由进程切换来完成的。进程阻塞的步骤如下:

(1) 停止进程执行,将现场信息保存到 PCB;

(2) 修改进程 PCB 的有关内容,如进程状态由运行态改为等待态等,并把状态已修改的进程移入相应事件的等待队列中;

(3) 转入进程调度程序,调度其他进程运行。

进程唤醒的步骤如下:

(1) 从相应的等待队列中移出进程;

(2) 修改进程 PCB 的有关内容,如进程状态改为就绪态,并将进程移入就绪队列;

(3) 若被唤醒的进程比当前运行进程的优先级高,重新设置调度标志。

阻塞原语和唤醒原语的作用正好相反,如果进程因某种事件调用阻塞原语来阻塞自己,则等待事件发生后,必须由与其相关的另一进程调用唤醒原语来唤醒被阻塞进程,否则,被

阻塞进程会因未被唤醒而处于永远阻塞状态。

4）进程的挂起与激活

当出现引起挂起的事件时，系统或进程会利用挂起原语把指定进程或处于等待态的进程挂起，其执行过程大致是：检查要被挂起的进程状态，若处于活动就绪态，就修改为挂起就绪态；若处于等待态，就修改为挂起等待态，被挂起进程 PCB 的非常驻部分要交换到磁盘兑换区。

当系统资源尤其是主存资源充裕或请求激活指定进程时，系统或相关进程会调用激活原语把指定进程激活，此原语所做的工作是：把被挂起进程 PCB 的非常驻部分调入主存，然后修改其状态，挂起等待态改为等待态，挂起就绪态改为就绪态，并将进程移入相应的队列中。需要注意的是，挂起原语既可由进程自己也可由其他进程调用，但是激活原语只能由其他进程调用。

6.3 存储保护

对于一个安全操作系统，存储保护是最基本的要求，存储保护指的是内存保护，由于任何程序和数据必须占用内存空间才能得以执行和处理，因此内存储器是计算机中的共享资源，即使对于单用户单任务的个人计算机，内存也是被用户程序和操作系统所共享的，在多道环境下更是被多个进程所共享。为了防止共享失去控制和产生不安全问题，对内存进行保护是必要的。

存储保护主要是指保护用户在存储器中的程序和数据，对于单用户单任务系统，内存中一次装入一个进程，存储保护机制应该防止用户程序对操作系统的影响，在支持多道程序并发运行的多任务系统中，还要进一步要求存储保护机制对各用户进程的存储区域实行互相隔离。

存储保护的主要目的如下：

(1) 防止对内存的未授权访问。

(2) 防止对内存的错误读写，如向只读单元写。

(3) 防止用户的不当操作破坏内存数据区、程序区或系统区。

(4) 多道程序环境下，防止不同用户的内存区域相互影响。

(5) 将用户与内存隔离，不让用户知道数据或程序在内存中的具体位置。

常用的内存保护技术有单用户内存保护、多道程序的保护、分段和分页保护技术。要了解各种存储保护技术，首先简要了解操作系统的地址转换技术。

6.3.1 地址转换

大多数应用程序、操作系统和实用程序都用高级程序设计语言或汇编语言编写，所编写的程序称为源程序，源程序中的符号名集合所限定的空间称为程序名字空间。源程序是不能被计算机直接运行的，需要通过如图 6.6 所示的三个阶段的处理后才能装入主存运行。

1. 编译

源程序经过编译程序(Compiler)或汇编程序(Assembly)的处理生成目标代码。一个

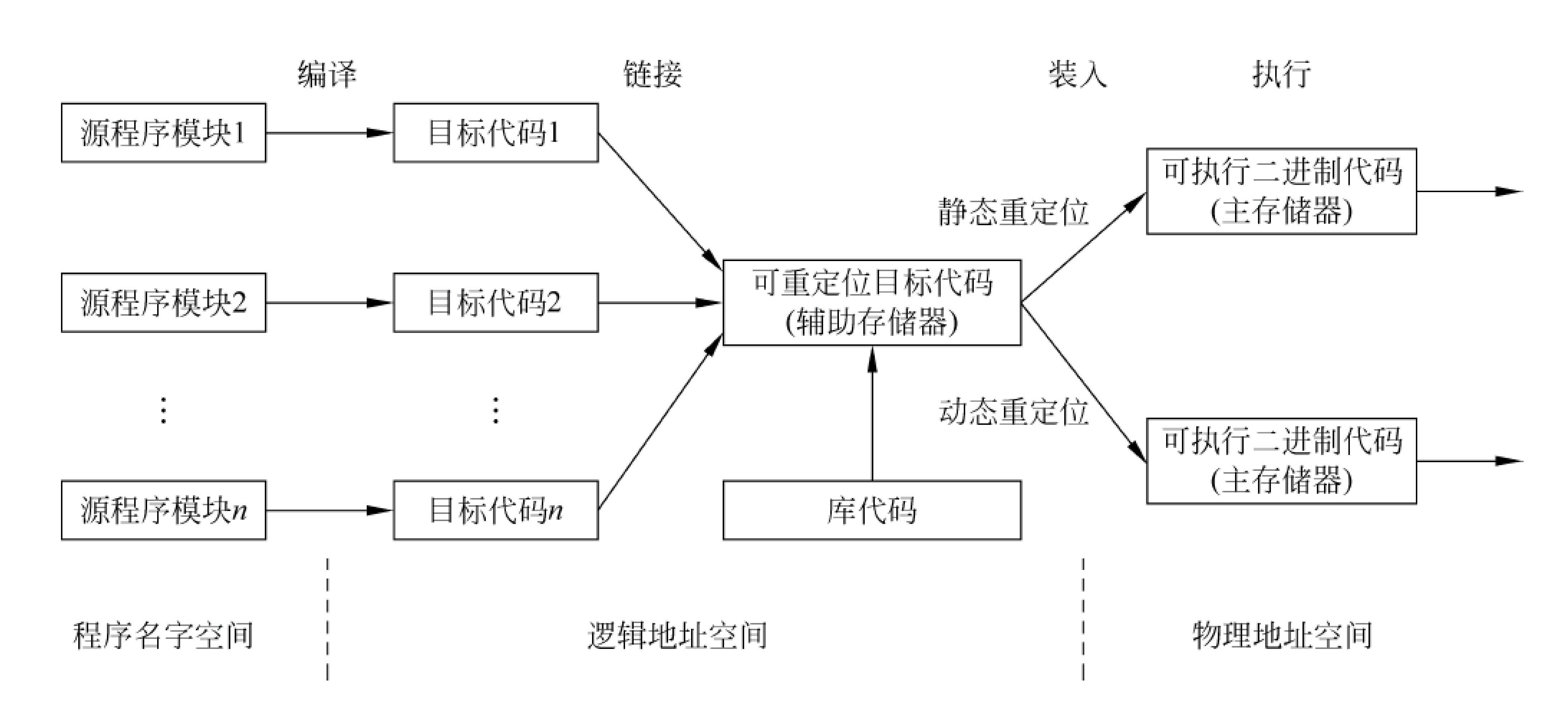


图 6.6　源程序执行过程

程序可由独立编写且具有不同功能的多个源程序模块组成，在 C 程序设计模型中，至少分为三个程序模块：文本段、数据段和堆栈段。由于模块中包含外部引用，即指向其他模块中的数据或指令的操作数地址，或包含对库函数的引用，编译程序负责记录引用的发生位置，编译或汇编的结果将产生相应的多个目标模块，每个目标模块都附有供引用的内部符号表和外部符号表。符号表中依次给出各个符号名及在本目标模块中的名字空间地址，外部符号在模块被链接时进行转换。例如，编写一个名为 simplecomputing 的源程序，其主程序 main 中有求平方根库函数 SQRT 和子程序调用指令 SUB1，SQRT 是函数库中已被编译成可链接的目标模块的标准子程序，SUB1 是另一个模块中定义的已被编译成可链接的子程序，这时所调用的入口地址均是未知的；编译程序或汇编程序将在外部符号表中记录外部符号名 SQRT 和 SUB1，同时两条调用指令指向函数和子程序的位置。

2. 链接

链接程序(Linker)的作用是把多个目标模块链接成一个完整的可重定位程序(其中包括应用程序要调用的标准库函数、所引用的其他模块中的子程序)，需要解析内部和外部符号表，把对符号名的引用转换为数值引用，将要涉及名字地址的程序入口点和数据引用点转换为数值地址。仍采用上例，Linker 首先将主程序调入工作区，然后，扫描外部符号表，获得外部符号名 SQRT，用此名字从标准函数库中找出函数的 sqrt. o 并装入工作区，拼接在主程序的下面；SQRT 函数的主存位置就是调用 SQRT 指令的入口地址，将此指令代真；调用 SUB1 的链接过程与此相似，只是从另一个模块中找到 sub1. o 的位置并进行指令代真，经过链接处理后，主程序 main 和 SQRT 函数和 SUB1 子程序链接成完整的可重定位目标程序 simplecomputing. o。

可重定位目标程序又称装载代码模块，它存放于磁盘中，由于程序在主存中的位置不可预知，链接时程序地址空间中的地址总是相对某个基准(通常为 0)开始编号的顺序地址，称为逻辑地址或相对地址。

3. 装入

在加载一个装载代码模块之前，存储管理程序总会先分配一块实际主存区给进程，装入

程序(Loader)根据指定的主存区首地址,将进程装入内存。磁盘中的装载代码模块所使用的是逻辑地址,其逻辑地址集合称为进程的逻辑地址空间。逻辑地址空间可以是一维的,这时逻辑地址限制在从0开始顺序排列的地址空间内;逻辑地址空间也可以是二维的,这时整个程序被分为若干段,每段都有不同的段号,段内地址从0开始顺序编址。进程运行时,其装载代码模块将被装入物理地址空间中,此时程序和数据的实际地址通常不可能同原来的逻辑地址一致。物理主存储器从统一的基地址开始为存储单元顺序编址,称为物理地址或绝对地址,物理地址的总体构成物理地址空间。需要注意的是,物理地址空间是由存储器地址总线扫描出来的空间,其大小取决于实际安装的主存容量。为了使进程能正确运行,需要把逻辑地址转换(绑定)为物理地址,这个过程称为地址重定位、地址映射或地址转换,地址重定位有以下两种方式。

1)静态地址重定位

由装入程序实现装载代码模块的加载和地址转换,把它装入分配给进程的主存指定区域,其中的所有逻辑地址修改成主存物理地址,称为静态重定位(Static Relocating Address)。地址转换工作在进程执行前一次完成,无须硬件支持,易于实现,但不允许程序在执行过程中移动位置。这种技术只在早期单用户单任务系统中使用过。

2)动态地址重定位

由装入程序实现装载代码模块的加载,把它装入分配给进程的主存储指定区域,但对链接程序处理过的应用程序的逻辑地址则不做任何修改,程序主存起始地址被置入硬件专用寄存器——重定位寄存器,如图6.7所示。程序执行过程中,每当CPU引用主存地址(访问程序和数据)时,由硬件截取此逻辑地址,并在它被发送到主存储器之前加上重定位寄存器的值,以便实现地址转换,称为动态重定位(Dynamic Relocating Address),地址转换推迟到最后的可能时刻,即进程执行时才完成。与静态地址重定位相比,动态地址重定位具有允许程序在主存中移动、便于程序共享和主存利用率高等优点。

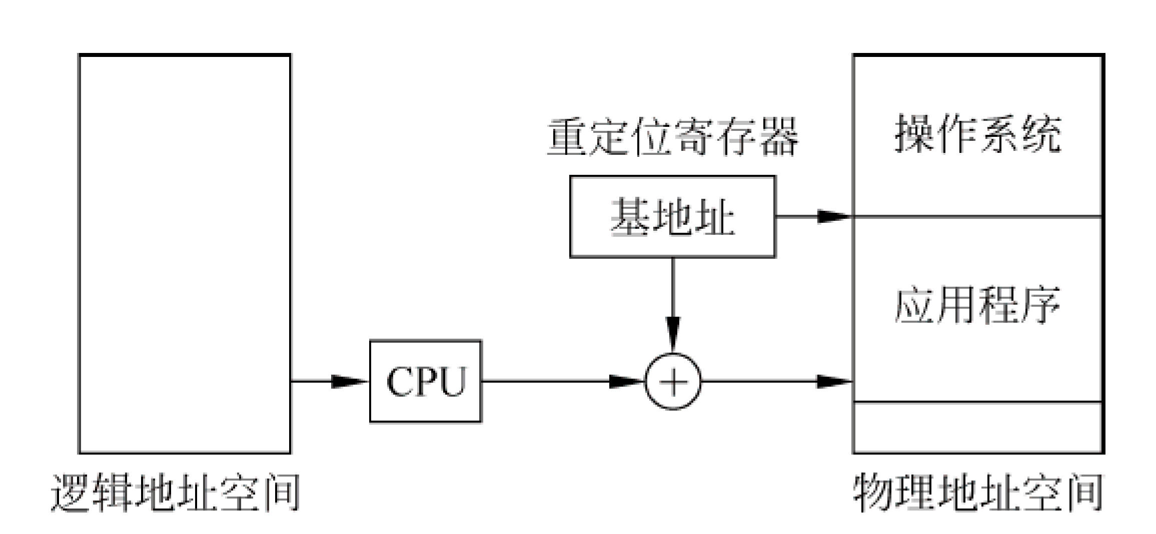


图6.7 动态地址重定位

在多道程序系统中,可用的主存空间常常被许多进程共享,程序员编程时不可能事先知道程序执行时的物理驻留位置,而且必须允许程序因对换或空闲区收集而被移动,这些现象都需要程序的动态地址重定位,即允许正在执行的程序在不同时刻处于主存储器的不同位置。从系统效率出发,动态地址重定位要借助于硬件地址转换机制来实现,重定位寄存器的内容通常保存在进程控制块中,每当执行进程上下文切换时,当前运行进程的重定位寄存器中的内容与其他相关信息一起被保护起来,新进程的重定位寄存器的内容会被恢复,这样进程就在上次中断的位置恢复运行,所使用的是与上次在此位置的相同的主存基地址。

6.3.2 存储保护方式

存储保护主要涉及防止地址越界和控制信息正确存取。计算机系统中存在操作系统和多个应用程序,它们在主存储器中各有自己的存储区域,操作系统必须对主存储区中的程序和数据进行保护,以确保各道程序只能访问自己的主存储区而不能相互干扰,从而避免受到其他程序有意或无意的破坏。无论采用何种地址重定位方式,通常进程运行时所产生的所有主存访问地址都应进行检查,确保进程仅访问自己的主存区,这就是地址越界保护。地址越界保护依赖于硬件设施,常用的有界地址和存储键。如何保证存取的正确性呢?进程在访问主存时,操作系统要对当前进程对内存区域的访问权限进行检查,如允许读、写、执行等,从而确保数据的安全性和完整性,防止有意或无意的误操作破坏主存信息,这就是信息存取保护。

存储保护方式跟具体的存储管理机制关联,支持多道程序设计的存储管理技术先后出现了固定分区存储管理、可变分区存储管理、分页存储管理、分段存储管理等技术,下面依次介绍。

1. 分区式存储管理机制及内存保护

分区式存储管理方式又分为固定分区和可变分区存储管理方式。固定分区存储管理的基本思想是:主存空间预先被划分成数目固定不变的分区,各分区的大小不等,每个分区只装一个作业,若多个分区中都装有作业,则它们可以并发执行,这是支持多道程序设计的最简单的存储管理技术。早期的 IBM 操作系统 OS/MFT(Multiprogramming with a Fixed Number of Tasks)就采用这种存储管理技术。

固定分区存储管理的缺点是由于预先规定了分区的大小,使得大作业无法装入,其次,主存空间的利用率不高,作业很少会正好填满分区。

可变分区(Variable Partition)存储管理按照作业的大小划分分区,但分区划分的时间、大小、位置都是动态的。系统把作业装入主存时,根据其所需要的主存容量查看是否有足够的空间,若有,则按需分割一个分区分配给此作业,若无,则令此作业等待主存资源。由于分区的大小是按照作业的实际需求量而定的,且分区的数目也是可变的,所以,可变分区能够克服固定分区中的主存资源的浪费,有利于多道程序设计,提高主存资源的利用率。使用可变分区管理的一个例子是 IBM 操作系统 OS/MVT(Multiprogramming with a Variable Number of Tasks)。

在可变分区中,进程装入的是一个连续的主存区域,随着进程不断地装入和撤销,导致主存中常常出现分散的小空闲区,称之为“碎片”。有时“碎片”会小到连小进程都容纳不下,这样不但浪费主存资源,还会限制进入主存的进程数目。当找不到足够大的空闲区来装入新进程时,可采用移动技术把已在主存中的进程分区连接到一起,使分散的空闲区汇集成片,这就是移动技术,采用移动技术虽然可以解决“碎片”的问题,但是处理器的开销太大。

在分区存储管理方式下,系统中有多个用户程序在内存中,每个进程只占用一个分区,不存在进程存储空间的交错,因此在存储保护时,可借助于基址寄存器存储当前执行进程所在分区的起始地址,借助于边界寄存器存储当前执行进程所在分区的上边界地址,进程运行时产生的所有地址都需检查是否在基址和上边界之间,若不在则报错,以此确保每个进程仅

能访问自己的分区,如图 6.8 所示。

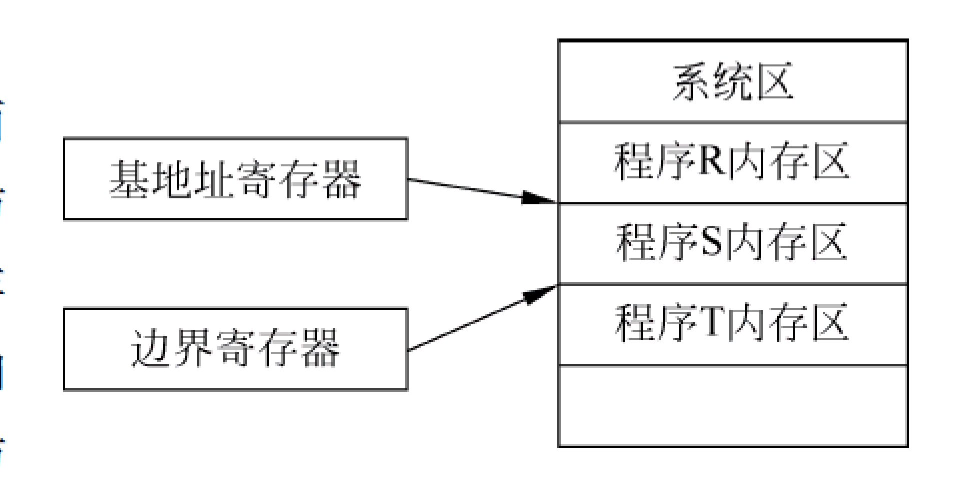


图 6.8 分区存储管理

用这种方法可以把程序完整地封闭在上下两个边界地址空间中,可以有效防止一个用户程序访问其他程序空间。如果使用多对基址和边界寄存器,还可以把用户的可读写数据区与只读数据区和程序区互相隔离,这种方法可以防止程序自身的访问错误。例如,可以防止向程序区或只读数据区的写访问。

2. 分页存储管理机制及内存保护

为了解决分区存储管理“碎片”问题,又产生了分页存储管理。采用分页存储管理允许程序放到若干不相邻的空闲块中,既可免除移动信息工作,又可充分利用主存空间,消除动态分区法中的“碎片”问题,从而提高主存空间的利用率。分页存储管理涉及的基本概念如下。

1) 页面

进程逻辑地址空间分成大小相等的区,每个区称为页面或页,页号从 0 开始依次编号。

2) 页框

把主存物理地址空间分成大小相等的区,其大小与页面大小相等,每个区是一个物理块或页框,块号从 0 开始依次编号。

3) 逻辑地址

分页存储器的逻辑地址由两部分组成:页号和页内偏移,格式如图 6.9 所示。

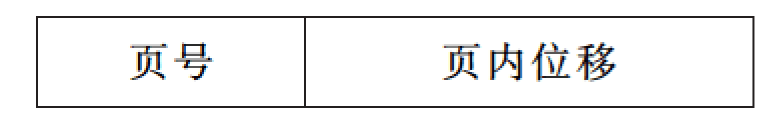


图 6.9 分页存储管理的逻辑地址

图 6.9 中的前半部分表示地址所在页面的编号,后半部分表示页内位移。计算机地址总线通常是 32 位,页面尺寸若规定为 12 位(页长 4KB),那么,页号共 20 位,表示地址空间最多可包含 2^{20} 个页面。

采用分页存储管理,逻辑地址是连续的,用户在编制程序时仍使用相对地址,不必考虑如何分页,由硬件地址转换机构和操作系统的管理需要来决定页面尺寸,从而确定主存分块大小。进程在主存中的每个页框内的地址是连续的,但页框之间的地址可以不连续,进程主存地址由连续到离散的变化为虚拟存储器的实现奠定了基础。

4) 页面和地址转换

在进行存储分配时,以页框为单位,进程的信息有多少页,那么,把它装入主存时就分配多少块,虽然进程的逻辑地址分成页面后是连续的,但是被装入内存后的对应物理块(页框)未必紧邻,即进程的信息按页面分散存放在主存不相邻的页框中,那么,当进程的程序和数据被分散存放在主存中后,其页面与被分配的页框如何建立联系呢?逻辑地址(页面)如何转换成物理地址(页框)呢?进程被装入后的物理地址空间由连续变为离散后,如何保证程序正确执行呢?地址转换仍然采用动态重定位技术,让程序在执行时动态进行地址变换,由于程序以页面为单位存储,所以为每个页面设立一个重定位寄存器,这些重定位寄存器的集合称为页表(Page Table)。页表是操作系统为进程建立的,是程序页面和主存对应页框的对照表,页表中的每一栏指明程序中的一个页面和分得的页框之间的对应关系。使用页表的目的是把页面映射为页框,从数学角度而言,页表是一个函数,其变量是页面号,函数值是

页框号,通过页表可以把逻辑地址中的逻辑页面域替换成物理页框域。为了减小系统开销,通常不用硬件而是直接在主存中开辟存储区存放进程页表,系统另设置专用硬件——页表基址寄存器,存放运行进程的页表起始地址,以加快地址转化速度。系统应为主存中的进程进行存储分配,并建立页表,指出逻辑地址页号与主存页框号之间的对应关系,页表的长度随进程大小而定。

进程运行前由系统把它的页表地址送入页表基址寄存器,运行时借助于硬件的地址转换机构,按页面动态地址重定位,当 CPU 获得逻辑地址后,由硬件自动按设定的页面尺寸分成两部分:页号 p 和页内位移 d,先从页表基址寄存器找到页表基地址,再用页号 p 作为索引查找页表,得到对应的页框号,根据关系式:

$$物理地址=页框号\times块长+页内偏移$$

计算出欲访问的主存单元。因此,虽然进程存放在若干个不连续的页框中,但在执行过程中总能按正确的物理地址进行存取。

如图 6.10 所示是分页存储管理的地址转换,在实际进行地址转换时,只要把逻辑地址中的页内偏移 d 作为绝对地址中的低地址,根据页号 p 从页表中查得的页框号 b 作为绝对地址中的高地址,就组成访问主存储器的绝对地址。

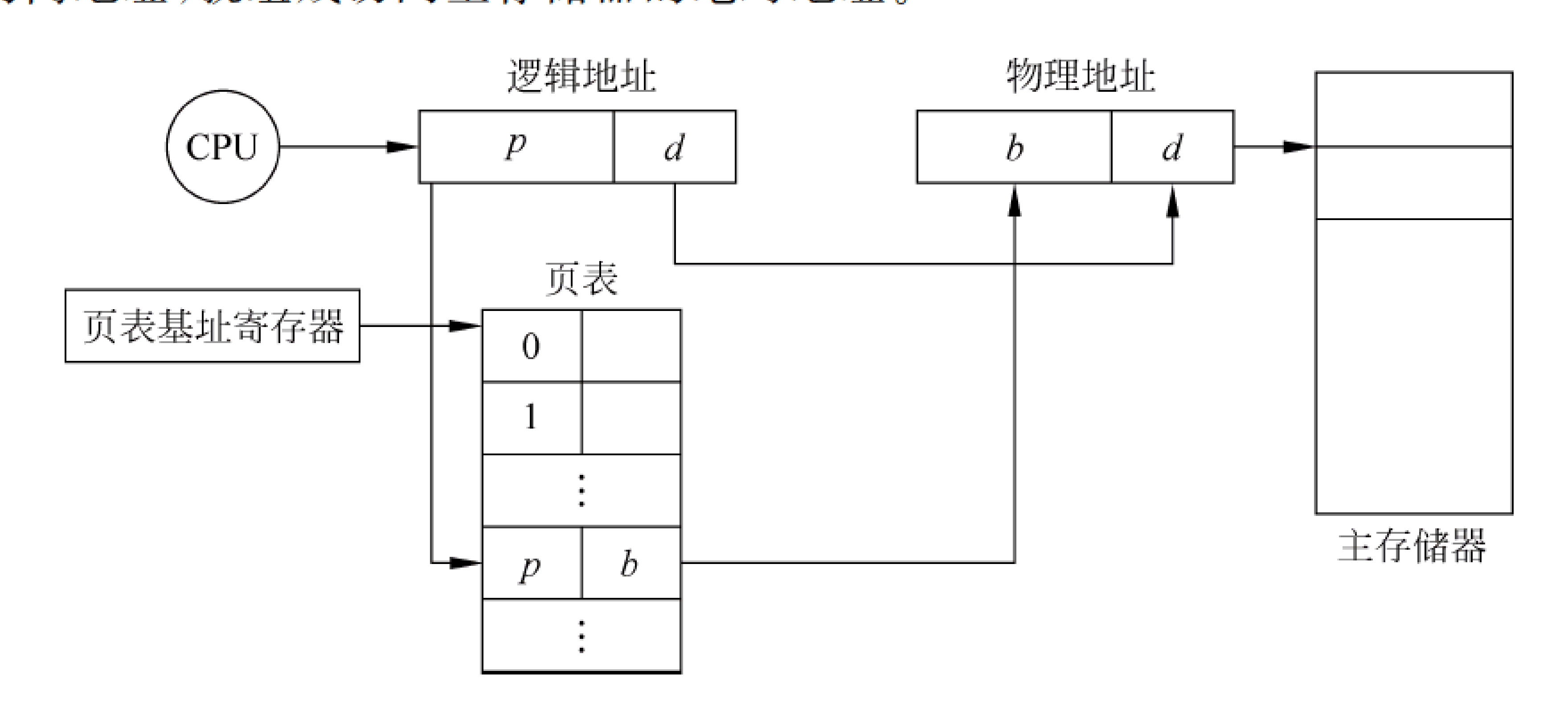


图 6.10　分页存储管理的地址转换

整个系统只有一个页表基址寄存器,只有占用 CPU 的进程才占有它。在多道程序中,当某道程序让出处理器时,应同时让出此寄存器供其他进程使用。

5）快表

页表可存放在寄存器或内存中,当页表存放在一组寄存器中时,地址转换只要从寄存器中取值就可得到页框号,转换速度快,但是硬件代价高;当页表存放在内存中时,可以降低系统开销,但是根据逻辑地址访问物理地址时,至少需要访问主存两次,一次访问页表,一次根据物理地址访问指令或数据,访问效率低。

为了提高运算速度,在硬件中设置相联存储器,用来存放进程最近访问的部分页表项,称为转换后援缓冲(Translation Lookaside Buffer,TLB)或快表,快表的存取时间远远小于主存,速度快但造价高,容量小,只能存放几十个页表项,快表项包括页号和对应的页框号,当把页号交给快表后,它通过并行匹配同时对所有快表项进行比较,如果找到,就立刻输出页框号,如果找不到,再查找主存中的页表,同时将找到的页号和页框号登记到快表中,当快表已满且要登记新页时,可采用“先进先出”等策略淘汰旧的快表项。

6）存储保护

在分页存储管理系统中，存储保护可以从两个方面实现，一个方面是在进行地址变换时，程序执行产生的页号应该小于页表长度，否则视为越界访问，有点类似于基址-限长存储保护；另一方面，可在页表中增加存取控制和存储保护的信息，通常的做法是在页表中增加标志位，指出此页的信息只读/读写/只可执行/不可访问等，当要访问某页时，先判断该页的存取控制和存储保护信息是否允许。例如，欲向只读页写入信息则指令停止执行。

3. 分段式存储管理机制及内存保护

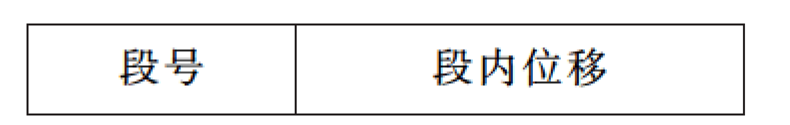
1）程序的分段结构

促使存储管理方式从固定分区到可变分区，从分区方式向分页方式发展的主要原因是为了提高存储空间利用率，而分段存储管理的引入则主要是满足用户（程序员）编程和使用上的要求，这些要求对于其他存储管理技术来说难以满足。在分页存储管理中，经链接编辑处理得到一维地址结构的可装配目标模块，这是从0开始编址的单一连续逻辑地址空间，虽然可以把程序划分成页面，但页面与源程序并不存在逻辑关系，也就难以对源程序以模块为单位进行分配、共享和保护。事实上，程序更多是采用分段结构，高级语言往往采用模块化程序设计方法。程序由若干程序段（模块）组成，例如，由主程序段、子程序段、数据段和工作区段组成，每段都从0开始编址，有各自的名字和长度，且实现不同的功能。

在源程序中，可用符号形式（指出段名和入口）调用某段功能，源程序经编译或汇编后，仍按照自身逻辑关系分为若干段，每段有一个段号，段之间的地址不一定连续，而段内地址是连续的。可见这是二维地址结构，模块化的程序被装入物理地址空间后，仍保持二维地址结构，这种地址结构需要编译程序的支持，但对程序员而言是透明的。

2）分段存储管理的基本原理

分段存储管理把进程的逻辑地址空间分成多段，提供二维逻辑地址，如图6.11所示。



段号	段内位移

图6.11 分段存储管理的逻辑地址

在分页存储管理中，页的划分，即逻辑地址划分成页号页内位移，是用户不可见的，连续的地址空间将根据页面的大小自动分页；而在分段存储管理中，地址结构是用户可见的，用户知道逻辑地址如何划分成段和段内位移，在设计程序时，段的最大长度由地址结构规定，程序中所允许的最多段数会受到限制，例如，PDP-11/45的段地址结构为：段号占3位，段内位移占13位，一个作业最多分为8段，各段长度达8KB。

分段存储管理的实现基于可变分区存储管理的原理。可变分区以整个作业为单位来划分和连续存放，也就是说，作业在分区内是连续存放的，但独立作业之间不一定连续存放。而分段方法是以段为单位来划分和连续存放的，为作业的各段分配一个连续的主存空间，而各段之间不一定连续。在进行存储分配时，应为进入主存的作业建立段表，各段在主存中的情况可由段表来记录，它指出主存中各段的段号、段起始地址和段长度。在撤销进程时，回收所占用的主存空间，并清除此进程的段表。

段表项实际上起到了基址/限长寄存器的作用，进程运行时通过段表可将逻辑地址转换成物理地址，由于每个用户作业都有自己的段表，地址转换应按各自的段表进行类似分页存储管理，也设置一个硬件——段表基址寄存器，用来存放当前占用处理器的作业段表的起始地址和长度，分段存储管理的地址转换和存储保护流程如图6.12所示，将段控制寄存器中

的段表长度与逻辑地址中的段号进行比较，若段号超过段表长度则触发越界中断，再利用段表项中的段长与逻辑地址中的段内位移进行比较，检查是否产生越界中断。

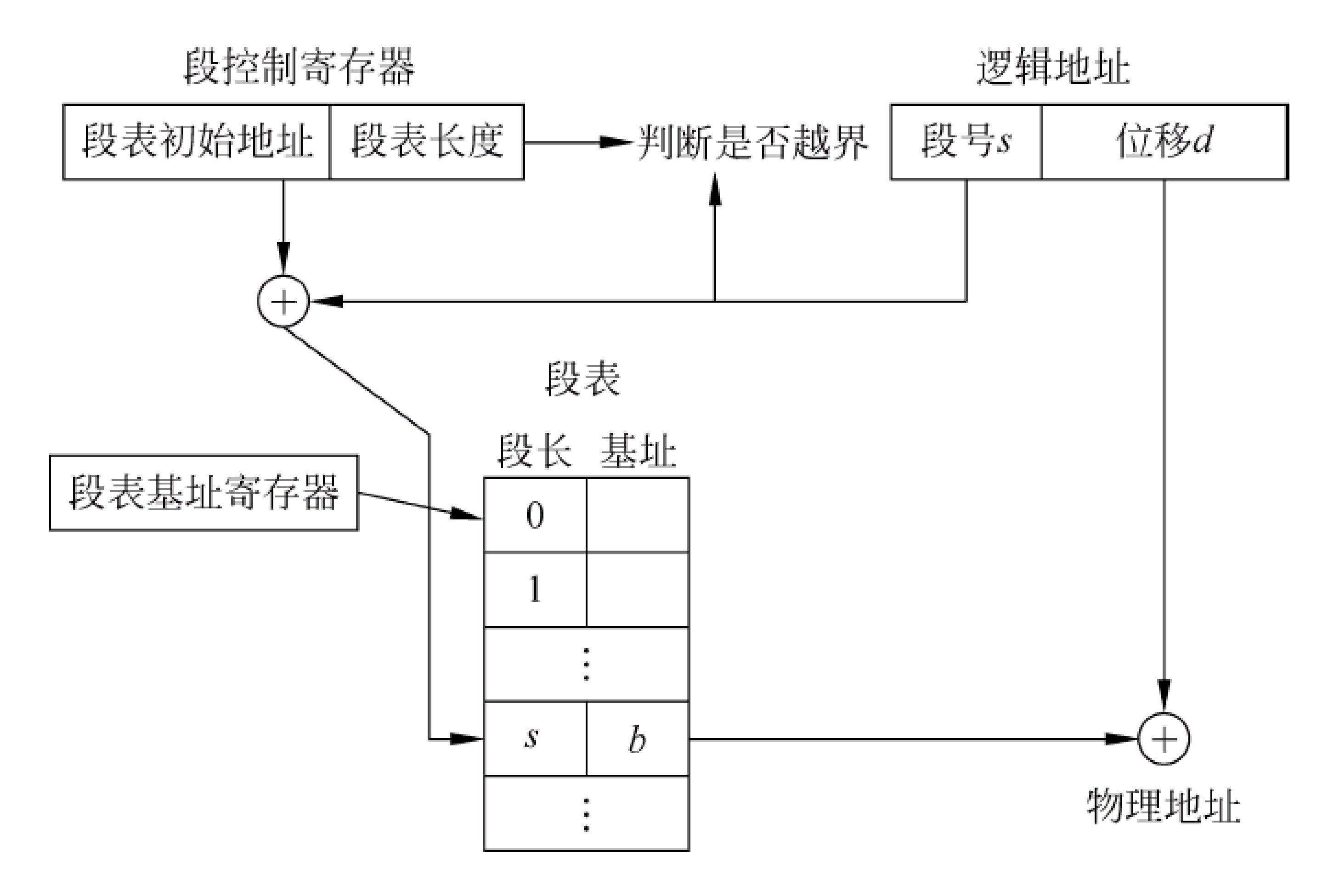


图 6.12　分段存储管理的地址转换

4. 虚拟存储管理

1）虚拟存储管理的概念

前面介绍的存储管理方式称为实存管理，只有将进程的全部信息装入主存后才能运行。而把进程的全部信息装入主存后，实际上并非同时使用，有些部分甚至从不使用，将进程在运行时不用的，或暂时不用的，或在某种条件下采用的程序和数据，全部驻留主存是对宝贵的存储资源的一种浪费，会降低主存利用率，这种做法很不合理。于是，计算机专家又提出新的想法：不必装入进程的全部信息，而仅将当前使用部分先装入主存，其余部分存放在磁盘上，待使用时由系统自动将其装进来，这就是虚拟存储管理技术的基本思路。当进程所访问的程序和数据在主存中时，可顺利执行，如果处理器所访问的程序和数据不在主存中，为了继续执行，由系统自动将这部分信息从磁盘装入，这叫做“部分装入”，如若此刻没有足够的空闲物理空间，便把主存中暂时不用的信息移至磁盘以腾出内存空间，这叫“部分替换”。如果“部分装入，部分替换”能够实现，那么，当主存空间小于进程的需要量时，进程也能运行；更进一步，当多个进程的总长超出主存总容量时，也可将进程全部装入主存，实现多道程序运行。这样，不仅能充分地利用主存空间，而且用户编程时不必考虑物理空间的实际容量，允许用户的逻辑地址空间大于主存物理地址空间，对于用户而言，好像计算机系统具有一个容量硕大的主存储器，称其为“虚拟存储器”（Virtual Memory）。

虚拟存储器可定义如下：在具有层次结构存储器的计算机系统中，自动实现部分装入和部分替换功能，能从逻辑上为用户提供一个比物理主存容量大得多的、可寻址的“主存储器”。实际上，虚拟存储器对用户隐蔽可用物理存储器的容量和操作细节，虚拟存储器的容量与物理主存大小无关，而受限于计算机的地址结构和可用的磁盘容量，如 Intel x86 的地址线是 32 位，则程序可寻址范围是 4GB。

如图 6.13 所示是虚拟存储器的概念图，其中，逻辑地址是从进程角度所看到的逻辑主存单元，而物理地址是从处理器角度所看到的物理主存单元，虚拟地址可以说是将逻辑地址映射到物理地址的一种手段。逻辑地址空间是程序员的编程空间，物理地址是程序的执行

空间,虚拟地址空间等同于实际物理主存加部分硬盘区域所组成的存储空间。

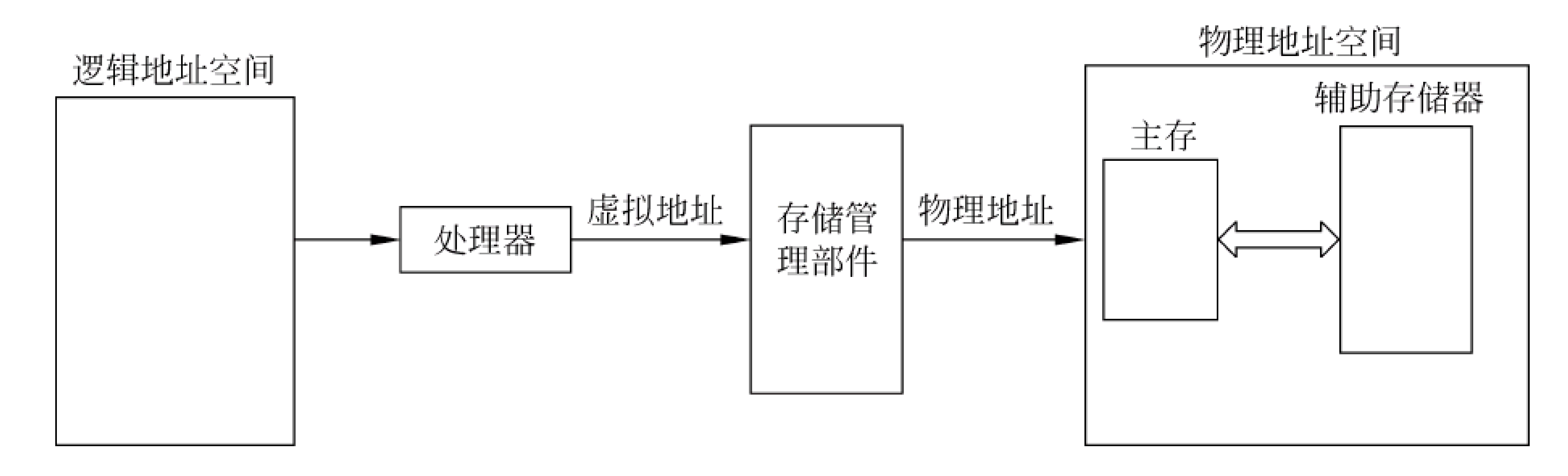


图 6.13 虚拟存储器

进程的信息不全部装入主存时能否保证程序正确运行呢?早在 1968 年,P. Denning 就开始着手研究程序运行时的局部性原理,发现程序和数据的访问都有聚集成群的倾向。某存储单元被使用之后,其相邻的存储单元也很快又被访问(称为空间局限性,Spatial Locality),或者最近访问过的程序代码和数据很快又被访问(称为时间局限性,Temporal Locality)。对程序的执行进行分析可以发现:第一,程序中只有少量分支和过程调用,大都是顺序执行的指令;第二,程序往往含有若干循环结构,由少量代码组成,而被多次执行;第三,过程调用的深度限制在小范围内,因而,指令引用通常被局限在少量的过程中;第四,许多计算涉及数组、记录之类的数据结构,对它们的连续引用是对位置相邻的数据项的操作;第五,程序中有些部分彼此互斥,不是每次运行时都用到,例如,出错处理部分在正常情况下用不到。种种情况说明,程序具有局部性,进程运行时没有必要把全部信息调入主存,只装入一部分进程信息的假设是合理的,此时,只要调度得当,不仅可正确运行进程,而且能在主存中放置更多进程,充分利用处理器和存储空间。虚拟存储器是基于程序局部性原理的一种假想的而非物理存在的存储器,其主要任务是基于程序局部性特点,当进程使用某部分地址空间时,保证将相应部分加载至主存中,这种将物理空间和逻辑空间分开编制、互相隔离,但又统一管理和使用的技术为用户编程提供了极大的方便。

虚拟存储器的思想早在 20 世纪 60 年代初就在英国 Atlas 计算机上出现,到 20 世纪 60 年代中期,较完整的虚拟存储器在分时系统 MULTICS 和 IBM 系列操作系统中得到实现,20 世纪 70 年代开始推广应用,逐步为广大计算机研制者和用户所接受。这项技术不仅用于大型计算机,也逐步用到微型计算机系统中。为了实现虚拟存储器,必须解决好以下问题:主存和辅存的统一管理问题、逻辑地址到物理地址的转换问题、部分装入和部分替换问题。目前,虚拟存储管理主要采用以下几种技术实现:请求分页、请求分段和请求段页虚拟存储管理。

2) 请求分页存储管理

(1) 请求分页虚拟存储管理的硬件支持

操作系统的存储管理依靠低层硬件的支撑来完成任务,此硬件称为主存管理部件(Memory Management Unit,MMU),它提供地址转换和存储保护功能,并支持虚拟存储管理和多任务管理。MMU 由一组集成电路芯片组成,逻辑地址作为输入,物理地址作为输出,直接送达总线,对主存单元进行寻址,其位置和功能如图 6.14 所示,其内部执行过程如图 6.15 所示。MMU 的主要功能列举如下。

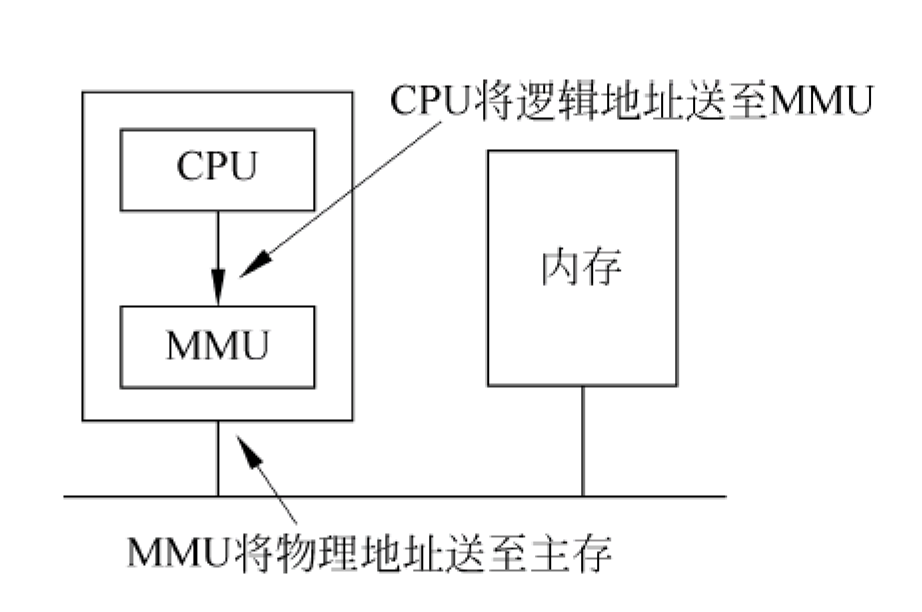


图 6.14 MMU 的位置和功能

CPU送入的逻辑地址

0010	000000000100

页号	页框号	在主存否
0	010	1
1	001	1
2	110	1
3	000	1
4	100	1
5	011	1
6	000	0
7	000	0
8	101	1
9	000	0
…		

110	000000000100

图 6.15 16 个 4KB 页面情况下 MMU 的内部操作

① 管理硬件页表基址寄存器

每当发生进程上下文切换时，系统负责把将要运行的进程的页表基地址装入硬件页表基址寄存器，此页表便成为活动页表，MMU 只对硬件页表基址寄存器所指出的活动页表进行操作。

② 分解逻辑地址

把逻辑地址分解为页面号和页内位移，以便进程地址转换。

③ 管理块表

对 TLB 进行管理，一是根据页号查找快表，找到相应的页框后去拼接物理地址；二是执行 TLB 的基本操作，即装入表目和清除表目，每次发生快表查找不命中的情况后，待缺页中断处理结束，把相应的页面和页框号装入。此外，每次写硬件页表基址寄存器时，负责清除快表项，将 TLB 清空。

④ 访问页表

当 TLB 不命中时，根据页表基址寄存器直接访问进程页表，若所需页面已装入，则可访问主存完成指令，同时，把此页面信息装入 TLB。

⑤ 发出相应中断

当查出页表中有页失效位或页面访问越界时，发出缺页中断或越界中断，并将控制权交给内核存储管理。

⑥ 管理特征位

负责设置和检查页表中的引用位、修改位、有效位和保护权限位等各个特征位。

下面考察 MMU 的工作过程。在图 6.15 中，给出一个逻辑地址 8196(二进制表示为

0010000000000100),MMU 进行映射,输入的 16 位逻辑地址被解释为 4 位页号和 12 位页内位移。用页号作为索引,找出虚页所对应的页框号,如果“在主存否”位为 1,表明此页在主存,把页框号复制到输出寄存器的高 3 位,再加上逻辑地址中的 12 位页内偏移,生成 15 位的物理地址(二进制表示为 110000000000100),并把它送到主存总线;如果“在主存否”位为 0,则将触发缺页中断,由操作系统进行缺页处理。

(2) 请求分页虚拟存储管理的基本原理

分页虚拟存储管理将进程信息的副本存放在辅助存储器上,当它被调度投入运行时,并不是把程序和数据一次性全部装入主存,而是仅装入当前使用的页面,进程执行过程中访问到不在主存中的页面时,再把所需信息动态地装入。目前常用的分页虚拟存储管理是请求分页(Demand Paging),采用这种存储管理方式时若需要执行不在主存的某条指令或使用某个数据时,系统会产生缺页(Page Default)中断,系统中断处理程序从磁盘中把此指令或数据所在的页面装入,这样做能够保证用不到的页面不会被装入主存。

请求分页存储管理与分页实存管理不同,仅将进程当前使用部分装入内存,其余部分存储在磁盘上,这势必会发生某些页面不在主存中的情况。那么,如何发现页面不在主存中呢?所采用的方法是:扩充页表项的内容,增加驻留标志位,又称页失效异常位,用来指出页面是否装入主存。当访问一个页面时,如果某页的驻留标志位为 1,表示此页已经在主存中,可被正常访问;如果某页的驻留标志位为 0,不能立即访问,产生缺页异常,操作系统根据磁盘地址将这个页面调入主存,然后重新启动相应的指令。那么,如何查找页面对应的磁盘地址呢?磁盘的物理地址由磁盘机号、柱面号、磁头号和扇区号所组成,通常规定扇区的长度等于页面的长度,页面与磁盘物理地址的对应表称为外页表,由操作系统管理。进程启动运行前系统为其建立外页表,并把进程程序页面装入辅助存储器,外页表也按进程号的顺序排列。为了节省主存空间,外页表可存放在磁盘中,当发生缺页中断需要查用时才被调入。

缺页中断是当发现当前访问页面不在主存时由硬件所产生的一种特殊中断信号,CPU 通常在一条指令执行完成后才检查是否有中断到达,也就是说只能在两条指令之间响应中断,但缺页中断却是在指令执行期间产生并获得系统处理的。而且,一条指令可能涉及多个页面,例如,指令本身跨页、指令所处理的数据跨页,完全有可能在执行一条指令的过程中发生多次缺页异常。

为了对页面实施保护和淘汰等各种控制,可在页表中增加标志位,如修改位、引用位和访问权限位等,用来跟踪页面使用情况。当页面被修改后,硬件自动设置修改位,当此页被调出主存时必须先被写回磁盘,引用位则在页面被引用(读或写)时设置,其值帮助系统进行页面淘汰,访问权限位则限定特定页面允许的访问权限(如读、写、执行等)。

可见,在请求分页虚拟存储管理中,页表中存放的是把逻辑地址转换成物理地址时硬件所需要的各类信息,其作用主要有:

① 获得页框号以实现虚实地址转换。

② 设置各种访问控制位,对页面信息进行保护。

③ 设置各种标志位来实现相应的控制功能(如缺页标志、修改标志、引用标志、锁定标志和淘汰标志等)。

下面讨论使用快表、页表存放在主存的情况下,请求分页虚实地址转换的过程。当进程

被调度到 CPU 上运行时，操作系统自动把此进程 PCB 中的页表起始地址装入硬件页表基址寄存器中，此后，进程开始运行并要访问某个虚地址，MMU 工作，它将完成虚线框内的任务，如图 6.16 所示。

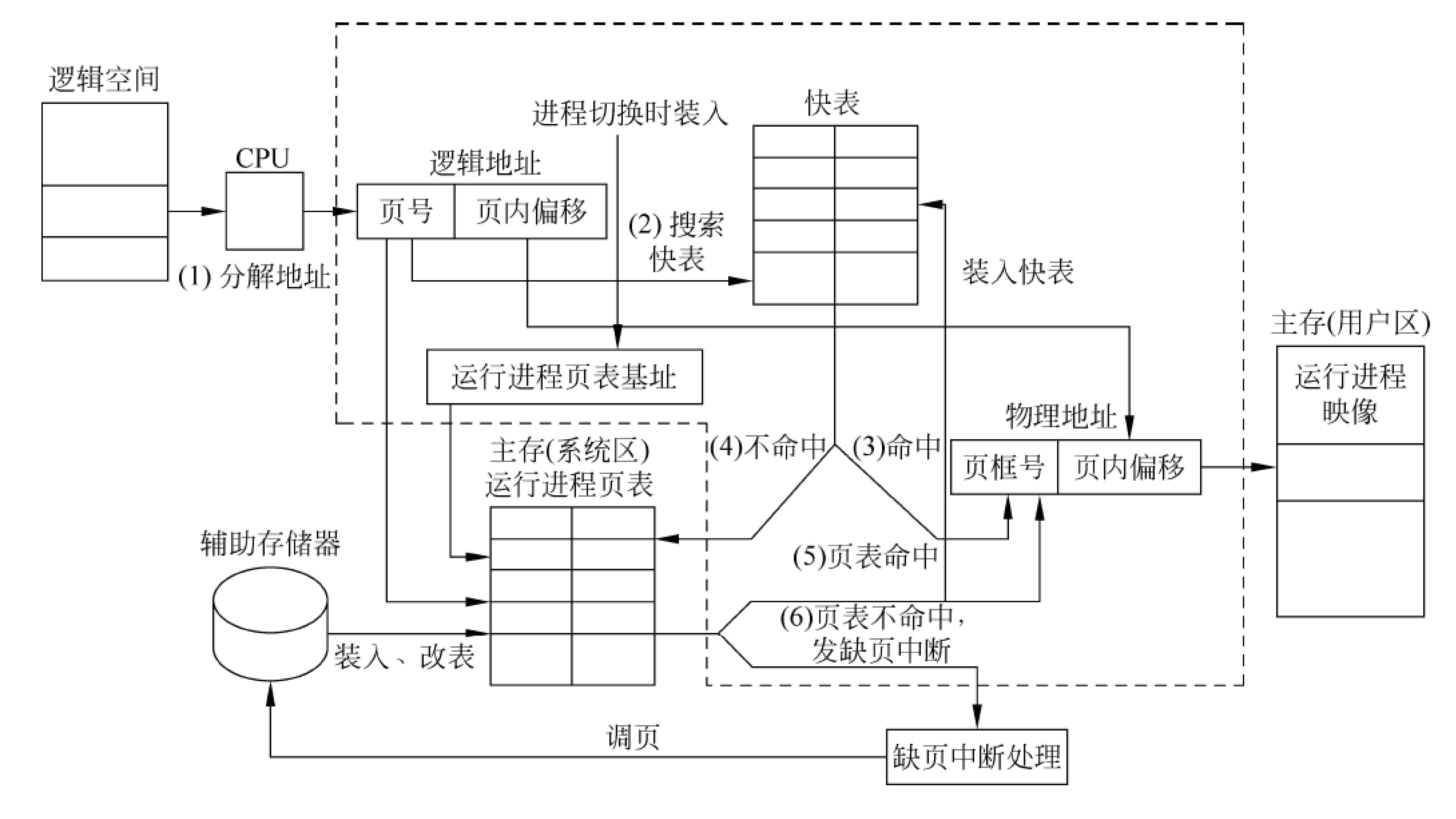


图 6.16　请求分页虚实地址转换

① MMU 接收 CPU 传送过来的逻辑地址并自动按页面大小把它从某位起分解成两部分：页号和页内位移。

② 以页号为索引搜索快表 TLB。

③ 如果命中，立即送出页框号，并与页内位移拼接成物理地址，然后进行访问权限的检查，如获通过，进程就可以访问物理地址。

④ 如果不命中，由硬件以页号为索引搜索进程页表，页表基址由硬件页表基址寄存器指出。

⑤ 如果在页表中找到此页面，说明所访问页面已在主存中，可送出页框号，并与页内位移拼接成物理地址，然后进行访问权限检查，如获通过，进程就可以访问物理地址，同时要把这个页面的信息装入快表 TLB，以备再次访问。

⑥ 如果发现页表中的对应页面失效，MMU 发出缺页中断，请求操作系统进行处理，MMU 工作到此结束。

在分页虚拟存储系统中，由于页面在需要时是根据进程的请求装入主存的，因此称为请求分页虚拟存储管理，这种技术的优点是：进程的程序和数据可按页分散存放在主存中，既有利于提高主存利用率，又有利于多道程序运行。这种技术的缺点是：要有一定的硬件支持，要进行缺页中断处理，机器成本增加，系统开销加大，此外，页内会出现碎片，如果页面较大，则主存的损失仍然很大。

3）请求分段存储管理

请求分段存储管理也为用户提供比主存实际容量大得多的虚拟主存空间。请求分段虚拟存储系统把作业的所有分段的副本都存放在辅助存储器中，当作业被调度投入运行时，首

先把当前需要的段装入主存,在执行过程中访问到不在主存的段时再将其动态装入。因此段表中必须增设供管理使用的若干表项,如特征位、存取权限、扩充位、标志位、主存起始地址和段长等。其中,特征位指示段是否在主存中,存期权限给出段的可访问模式,扩充位指出段可否动态扩充,标志位又分为修改位、访问位、可否移动位等。

有关请求分段存储管理的地址转换和存储保护这里不作详述。

6.4 用户身份认证

身份认证是操作系统提供的第一道安全防线,可以防止非授权用户登录系统,在操作系统中,身份认证一般是在用户登录时发生,系统提示用户输入口令,然后判断输入的口令与系统中存储的该用户的口令是否一致,这种口令机制是简便易行的鉴别手段,但比较脆弱,较为安全的身份认证机制如一次性口令机制、生物认证等已经取得长足进展,逐步达到了实用阶段,有关身份认证的方式在第 3 章中已经作了详细介绍,这里不再赘述。

6.5 访 问 控 制

在操作系统领域中,访问控制一般涉及自主访问控制、强制访问控制、基于角色的访问控制三种形式。有关访问控制的基本原理和实现方式的内容在第 4 章已经作了详细介绍,这里不再赘述。

6.6 审　　计

6.6.1 审计的概念

审计是模拟社会监督机制而引入计算机系统中的,是指对系统中安全相关的活动进行记录、检查及审核。审计的主要目的是检测非法用户对计算机系统的入侵行为以及合法用户的误操作。审计能够完整记录涉及系统安全的操作行为,是一种确保系统安全的事后追查手段。审计为系统进行事故原因的查询、定位,事故发生前的预测、报警以及事故发生之后的实时处理提供了详细、可靠的依据和支持。

审计是操作系统安全的一个重要方面,美国国防部的橘皮书中就明确要求“可信计算机必须向授权人员提供一种能力,以便对访问、生成或泄露秘密或敏感信息的任何活动进行审计,根据一个特定机制或特定应用的审计要求,可以有选择地获取审计数据。但审计数据中必须有足够细的粒度,以支持对一个特定个体已发生的动作或代表该个体发生的动作进行追踪”。在我国 GB 17859-1999 中也有相应的要求。

审计通常和报警功能结合起来,每当有违反系统安全的事件发生或者有涉及系统安全的重要操作进行时,就及时向安全操作员终端发送相应的报警信息。审计也是入侵检测系统、数字取证、网络安全管理等系统的基本构件之一。

审计过程一般是一个独立的过程，它与系统其他功能相隔离，同时要求操作系统必须能够生成、维护及保护审计过程，使其免遭修改、非法访问及毁坏，特别要保护审计数据，要严格限制未经授权的用户访问。

6.6.2 审计事件

系统将所有要求审计或可以审计的用户动作都归纳成一个个可区分、可识别、可标志的审计单位，称为审计事件。审计事件是系统审计用户操作的最基本单位。

例如为了记录创建文件这一事件，系统可以设置标记为 create 的审计事件，当用户通过系统调用 create(“file1”，mode)或 open(“file1”，O_CREATE，mode)创建文件时，内核就会将该事件记录下来。

多数操作系统都在内核态和用户态记录审计事件，审计事件通常包括系统事件、登录事件、资源访问、特权使用、账号管理和策略更改等类别。表 6.1 给出了具体描述。

表 6.1 审计事件

类别名称	包括的主要事件
系统事件	系统启动、关机、故障等
登录事件	成功登录、各类失败登录、当前登录等
资源访问	打开、关闭、修改资源等
操作	进程、句柄等的创建与终止，对外设的操作、程序的安装和删除等
特权使用	特权的分配、使用和注销等
账号管理	创建、删除用户或用户组，以及修改其属性
策略更改	审计、安全等策略的改变

审计机制一般对系统定义一个固定审计事件集，即必须审计事件的集合。系统审计员可以通过系统提供的工具自定义需要审计的事件，用户的行为一旦落入用户事件集或系统固定审计事件集中，系统就会将这一信息记录下来，否则系统将不对该事件进行审计。

审计过程会增大系统的开销(CPU 时间和存储空间)，如果设置的审计事件过多，势必使系统的性能相应地下降很多(例如响应时间、运行速度等)，所以在实际设置过程中，审计机制应是对用户在系统中行为的一种有选择的记载，要选择最主要的事件加以审计，不能设置太多的审计事件，以免过多影响系统性能。系统审计员可以通过设置事件标准，确定对系统中哪些用户或哪些事件进行审计，审计结果存放于审计日志文件中，审计结果可以按要求的报表形式打印出来。

6.6.3 审计记录和审计日志

审计记录是指当审计事件发生时，由审计系统用审计日志记录相关信息。一般应包括如下信息：事件的日期和时间、代表正在进行事件的主体的唯一标识符、事件类型、事件的成功与失败等。对于标志和鉴别事件，审计记录应该记录下事件发生的源地点(如终端标识符)。对于涉及客体操作的事件，审计记录应该包含客体名等信息。

审计日志是存放审计结果的二进制结构文件，每个文件包含多条记录，当审计日志长度超过一定大小，系统会按照预先设置好的路径和命名规则产生一个新的日志文件。

【例 6-2】 Windows XP 的审计系统及审计日志。

Windows XP 的审计系统由操作系统内部的安全参考监视器(Security Reference Monitor,SRM)、本地安全中心(Local Security Authority,LSA)和事件记录器(Event Logger)等模块组成。LSA 负责管理审计策略,在每次审计事件发生时,审计日志记录由 SRM 和 LSA 根据相应系统或应用的通知生成,记录先被传输到 LSA,经过 LSA 处理后转发给事件记录存储。Windows XP 的审计日志主要分为系统日志、安全日志和应用日志三类,分别记录有关操作系统事件、安全事件和应用事件的发生情况,审计记录包含的内容如表 6.2 所示。

表 6.2 Windows XP 的审计系统

字段名称	内容	范例
类型	本条记录内容的基本类型,主要包括错误、警告、信息或正确审核、失败审核	错误
日期	事件发生时的年、月、日	2015-8-20
时间	事件发生时的小时、分、秒	21:36:59
来源	通报事件的系统或应用程序	Service Control Manager
分类	事件分类	登录/注销
事件	事件编号	101
用户	事件涉及的用户	SYSTEM
计算机	事件涉及的计算机	My Computer

6.6.4 一般操作系统审计的实现

实现审计时,首先要解决的问题是如何保证所有安全相关事件都能被审计。我们知道在一般的多用户多进程操作系统(如 UNIX、Linux 等)中,用户程序与操作系统的唯一接口是系统调用,也就是说当用户请求系统服务时,必须经过系统调用。因此,在系统调用的总入口处(称作审计点)增加审计控制,就可以成功地审计系统调用,也就是成功地审计了系统中所有使用内核服务的事件。

系统中有一些特权命令也属于可审计事件,而通常一个特权命令需要使用多个系统调用,逐个审计所用到的系统调用,会使审计数据复杂而难于理解,审计员很难判断出命令使用情况。因此虽然系统调用的审计已经十分充分,特权命令的审计仍然必要。为了实现对特权命令的审计,可以在被审计的特权命令的每个可能的出口处增加一个新的系统调用,专门用于该命令的审计。当发生可审计事件时,审计点调用审计函数并向审计进程发消息,审计进程是一个守护进程,完成审计信息的缓冲、存储、归档工作,如图 6.17 所示。

一般情况下,审计在系统开机引导时就会自动开启,审计管理员可以随时关闭审计功能。审计功能被关闭后,任何用户的任何动作就不再处于审计系统的监视下,也不再记录任何审计信息。

系统在审计时,要将审计信息写入日志中,自然会给系统带来时间上的额外开销,影响系统的性能。为了将这种时间开销减少到最低程度,审计系统不必每次有一条记录时就立即写入审计日志文件中,可在系统中开辟一片审计缓冲区。系统在大多数情况下只需将审

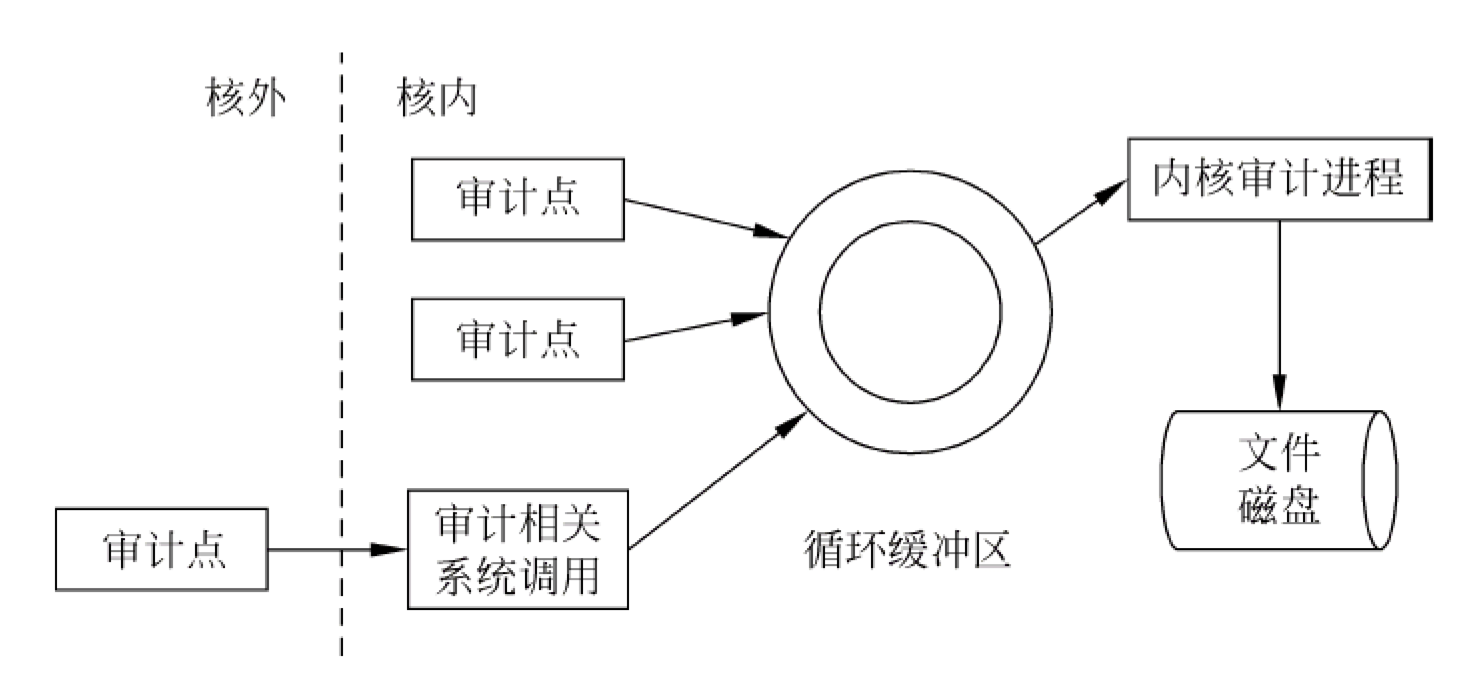


图 6.17　审计系统架构

计信息写入审计缓冲区中,只有在缓冲区已经写满或者内容达到一定的限度时,审计进程才一次性地将审计缓冲区中的有效内容全部写入日志文件中。

系统审计员可以根据需要选择审计信息,用文档或报告的形式打印出来,供各种分析需要,同时也可以将日志文件转储在除硬盘之外的存储媒体上,以节省系统磁盘空间。

6.7　最小特权管理

6.7.1　基本思想

在操作系统中,为了维护系统正常运行及其安全策略的实施,往往需要为某些特殊用户赋予一定的特权以直接执行一些受限的操作或进行超越安全控制策略的操作,例如执行软件安装、用户账户维护等。所谓特权是指超越访问控制的能力,它与访问控制相结合,可以提高系统的灵活性。目前主流操作系统中都存在一个无所不能的超级用户,Windows 的超级用户是 Administrator,UNIX、Linux 和 FreeBSD 的超级用户称为 ROOT,在这些系统中,超级用户拥有所有特权,可以执行任意操作,而普通用户不具有任何特权,这种特权管理方式有利于系统的维护和配置,却不利于系统的安全性。一旦超级用户的口令丢失或超级用户被冒充,将会对系统造成极大的损失。另外超级用户的误操作也使系统存在潜在安全隐患。因此,TCSEC 标准要求 B2 级以上的安全操作系统必须提供最小特权管理以确保系统安全。

最小特权的思想是系统不应赋予用户超过其执行任务所需特权以外的特权,或者说仅给用户赋予必不可少的特权,最小特权原则一方面赋予主体“必不可少”的特权以保证用户能完成承担的任务或操作,另一方面它仅给用户“必不可少”的特权从而能限制用户所能进行的操作。同时为了保证系统的安全性,不应对某个用户赋予一个以上的职责,而一般系统中的超级用户通常肩负系统管理、审计等多项职责,因而需要将超级用户的特权进行细粒度划分,分别授予不同的管理员,并使其只具有完成其任务所需的特权,从而减少由于特权用户口令丢失或错误软件、恶意软件、误操作所引起的损失。

社会流行的管理模式通常源于 18 世纪法国思想家孟德斯鸠提出的三权分立理论,即立法机构、执行机构和监督机构三权鼎立、相互牵制。这种管理模式可以引入到操作系统的安全管理之中,使得系统中不再有超级用户,而是将所有特权分解成一组细粒度的特权子集,

定义成不同的角色，使系统中的任何用户(组)，只具有完成其必需功能的“最小特权”，任何用户的权限都不足以操纵整个系统且相互制约，这就避免了超级用户的误操作或其身份被假冒而带来的安全隐患。

利用三权分立的思想可在系统中定义系统安全管理员、审计管理员、系统管理员三个特权管理职责。系统管理员拥有系统管理特权集，管理与系统相关的资源，包括用户身份管理、系统资源配置、系统加载和启动、系统运行的异常处理以及支持管理本地和异地灾难备份与恢复等。安全管理员拥有安全管理特权集，是整个系统安全策略的制定者，负责对系统中的主体、客体进行统一标记，对主体进行授权，配置一致的安全策略，并确保标记、授权和安全策略的数据完整性。审计管理员拥有审计管理特权集，是系统的监督者，负责系统运行中审计记录的保存和读取。各管理员只具有完成其任务所需的最小特权，不同管理员之间相互协作、相互制约，任何一个用户都不能获得足够的权利来破坏系统安全策略。

6.7.2 POSIX 权能机制

权能是有效实现最小特权的机制，其主要思想是将超级用户的所有特权划分成多种权能，每种权能代表某种特权操作的权限，系统根据一定的策略赋予进程权能，并根据进程拥有的权能进行特权操作的访问控制。基于权能的最小特权控制最早是由 Dennis 提出的，早期的安全系统允许进程本身携带一组对特定客体的访问权，并且在允许的情况下，一个进程可以在任何时候放弃或收回它的一些权能。POSIX 权能机制与传统的权能机制类似，但为系统提供了更为便利的权能管理和控制。

POSIX 标准将超级用户的权力细分成 26 个权能，如表 6.3 所示。

表 6.3 POSIX 权能

权　能	说　明
CAP_OWNER	拥有该权能可以改变文件的属主或属组
CAP_AUDIT	拥有该权能可以操作安全审计机制，包括写审计记录等
CAP_COMPAT	拥有该权能可以超越限制隐蔽通道所做的特别约束
CAP_DACREAD	拥有该权能可以超越自主访问控制的读检查
CAP_DACWRITE	拥有该权能可以超越自主访问控制的写检查
CAP_DEV	当设备处于私有状态时设置或获取设备安全属性以改变设备级别并访问设备
CAP_FILESYS	对文件系统进行特权操作，包括创建与目录的连接、设置有效根目录、制作特别文件
CAP_MACREAD	拥有该权能可以超越强制访问控制的读检查
CAP_MACWRITE	拥有该权能可以超越强制访问控制的写检查
CAP_MOUNT	拥有该权能可以安装或卸载一个文件系统
CAP_MULTIDIR	拥有该权能可以创建多级目录
CAP_SETPLEVEL	拥有该权能可以改变进程的安全级(包括当前进程本身的安全级)
CAP_SETSPRIV	拥有该权能可以超越访问和所有权限制；管理权能，用于给文件设置可继承和固定特权
CAP_SETUID	拥有该权能可以设置进程的真实、有效用户/组标识符
CAP_SYSOPS	拥有该权能可以完成几个非关键安全性的操作，包括配置进程记账、维护系统时钟、提高或设置其他进程的优先级、设置进程调度算法等

续表

权　能	说　明
CAP_SETUPRIV	用于非特权进程设置文件的权能状态，该权能不能超越访问和所有权限制
CAP_MACUPGRADE	允许进程升级文件安全级
CAP_FSYSRANGE	拥有该权能可以超越文件系统范围限制
CAP_SETFLEVEL	拥有该权能可以改变客体安全级
CAP_PLOCK	拥有该权能可以上锁一个内存中的进程，上锁共享内存段
CAP_CORE	拥有该权能可以转储特权进程、setuid 进程、setgid 进程的核心映像
CAP_LOADMOD	用于完成与可安装模块相关的可选择操作，如安装或删除内核可加载模块
CAP_SEC_OP	拥有该权能可以完成安全性有关的系统操作，包括配置可信路径的安全注意键、设置加密密钥等
CAP_DEV	拥有该权能可以对计算机设备进行管理，包括配置终端参数、串口参数、配置磁盘参数等
CAP_OP	拥有该权能可以进行开机、关机操作
CAP_NET_ADMIN	拥有该权能可以对计算机进行与网络有关的操作，包括可以使用 RAW、PACKET 端口号，可以绑定端口号低于 1024 端口，可以进行网卡接口配置、路由表配置等

在 POSIX 中定义了一个权能模型，在这个模型中，只有进程和可执行文件具有权能信息。进程的权能信息包含三组权能集，分别称为 permitted_cap、effective_cap 和 inheritable_cap（简记为 pP，pE，pI）。其中，permitted_cap 为允许权能集，表示进程拥有的最大权能集；effective_cap 为有效权能集，表示进程当前有效的权能集，系统根据进程的 effective_cap 进行访问控制（effective_cap 为 permitted_cap 的子集）；inheritable_cap 为继承权能集，表示进程可传递给其子进程的权能集。

可执行文件的权能信息也包含三组权能集，为了与进程的权能集名称相区别，分别称为 cap_forced、cap_effective 和 cap_allowed（分别简记为 fP、fE、fI）。其中，cap_allowed 表示文件被执行时可从原进程继承的权能集；cap_forced 表示文件被执行时强制赋给进程的权能集；cap_effective 则表示可执行文件开始执行时进程可以拥有的有效权能集。

新创建的子进程拥有和父进程相同的权能信息，而当子进程调用 exec()执行了新的程序时，系统将根据以下公式重新赋予子进程权能：

$pI' = pI$

$pP' = fP \mid (fI \,\&\, pI)$

$pE' = pP' \,\&\, fE$

其中，pI'、pP'、pE'为进程新的权能信息，pI、pP、pE 为进程原来的权能信息，fP、fI、fE 则为执行文件的权能信息。

根据以上公式，通过设置可执行文件的权能信息可以控制相应进程拥有的权能。新进程所能拥有的最大权能集由可执行文件的 fP、fI 以及原进程的 pI 共同决定，设置可执行文件的 fP 可以强行赋给进程权能，设置文件的 fI 则可继承进程原来的权能。

6.8 Windows 系统安全

Windows 系统是目前市场上占统治地位的操作系统，被广泛作为企业、政府部门以及个人计算机的系统平台，现在常用版本有 Windows XP、Windows 7 等。自 Windows 2000 以来，微软一直关注操作系统的安全设计和配置，提供了多种安全机制，了解 Windows 系统的安全机制，并制定精细的安全策略，用 Windows 构建一个高度安全的系统才能成为可能。

6.8.1 Windows 安全子系统的结构

Windows 系统基于经典的引用监控器模型来实现基本的对象安全模型，系统中所有主体对客体的访问都通过引用监控器作为中介，由引用监控器根据安全访问控制策略进行授权访问，所有访问记录也都由引用监控器生成审计日志。Windows 系统的安全子系统主要由本地安全授权子系统(LSASS)、安全引用监控器(SRM)、事件记录器(EventLog)等模块组成，如图 6.18 所示。

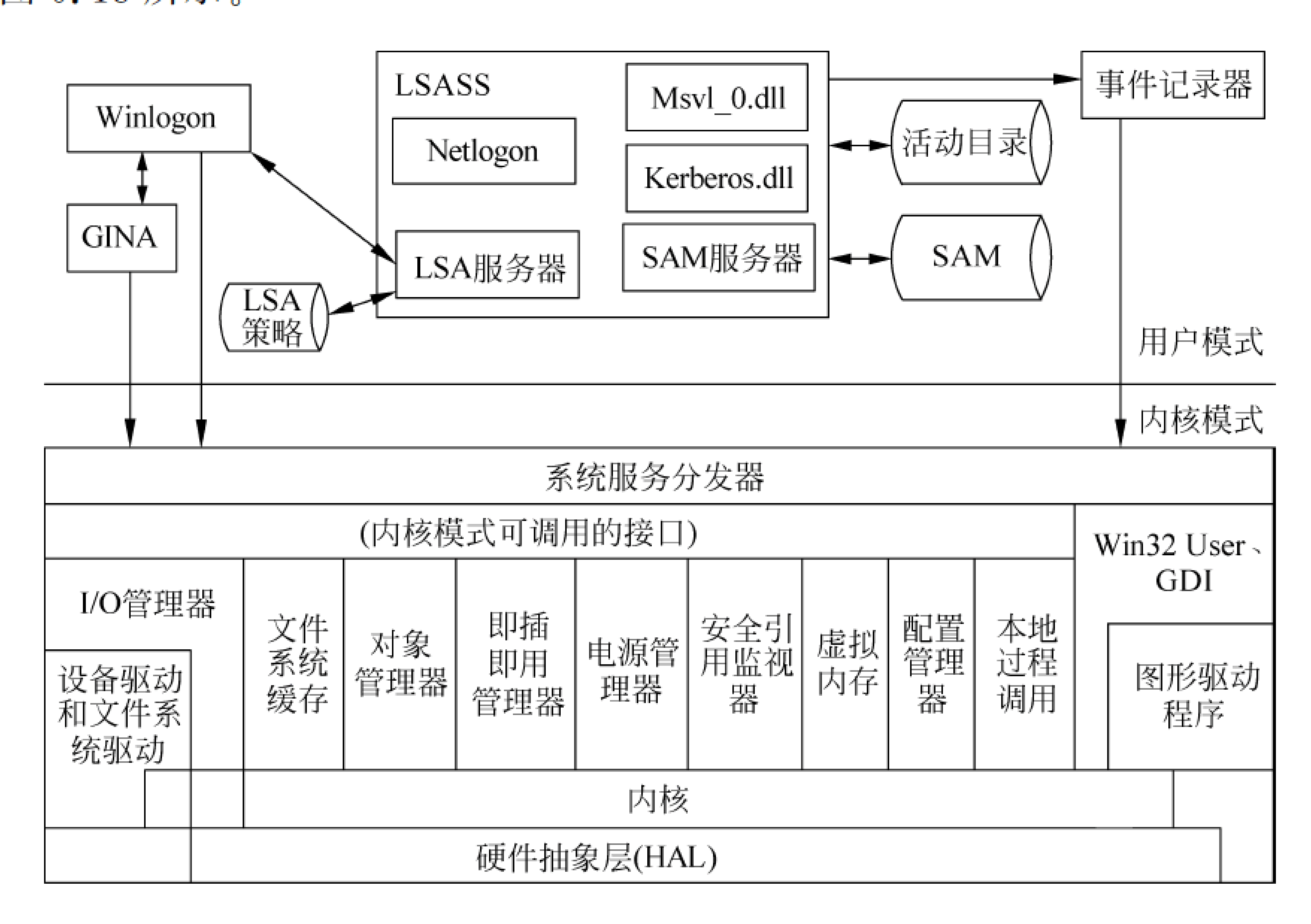


图 6.18 Windows 系统安全体系结构示意图

1. 身份认证

Winlogon 进程与 LSASS (Local Security Authority SubSyetm)中的 Netlogon 分别负责 Windows 本地和远程登录用户的身份认证，利用 LSASS 所提供的身份验证服务，来确定主体身份的真实性。这里主要讨论本地身份认证。

Winlogon 是 Windows 2000 以上版本提供本地交互式登录支持的一个组件，该进程对应的可执行文件为%SystemRoot%\System32\Winlogon.exe，主要负责管理与登录相关的安全工作。用户在 Windows 系统启动后按 Ctrl+Alt+Delete 组合键，会引起硬件中断，该中断信息被系统捕获后，操作系统即激活 Winlogon 进程。

Winlogon 调用图形化标识和认证(Microsoft Graphical Identification and Authentication, GINA)来显示“登录”对话框。GINA 是一个用户模式的 DLL,运行在 Winlogon 进程中,标准 GINA 是\Windows\System32\msgina.dll,负责显示用户信息输入界面,收集用户登录信息,并将用户登录信息通过安全信道反馈给 Winlogon。为了支持更多的交互登录验证方式,GINA 动态库是可以替换的,用户可以自己开发 GINA 动态库以实现其他身份验证方式,如智能卡、指纹、虹膜等。GINA 通过串接的方式来组合多种身份认证机制,例如自定义 GINA 的指纹识别模块串接到原本的 Windows XP 账号及密码认证之后。不过,这种方式会带来一个大问题,那就是当 GINA 串接前面的认证方式更新之后,有可能造成 GINA 串接断掉,让后面的认证进程失效。

在 Windows7 中使用了全新的凭据提供者 API 来取代原先的 GINA 机制。Windows 7 中可以同时挂接多个凭据提供者,这些凭据提供者之间以并联的方式组成,因此彼此之间不会有任何干扰。

Winlogon 在收集好用户的登录信息后,就调用本地安全授权(Local Security Authority,LSA)的 LsaLogonUser 命令,把用户的登录信息传递给 LSA,实际认证部分的功能是通过 LSA 来实现的,LSA 从 Winlogon 中获取用户的账号和密码,调用认证包,将用户信息加密处理后交给 SAM(Security Account Manager)服务器,SAM 服务器通过与存储在 SAM 数据库中的用户信息对比,以确定用户身份是否有效。Winlogon、GINA、LSA 三部分相互协作实现了 Windows 的本地登录认证功能,本地登录过程为:

(1) 用户首先按下 Ctrl+Alt+Del 组合键。

(2) Winlogon 检测到用户按下的组合键,就调用 GINA,由 GINA 显示登录对话框,以便用户输入账号和口令。

(3) 用户输入账号和口令,单击“确定”按钮后,GINA 把信息发送给 LSA 进行验证。

(4) 在用户登录到本机的情况下,LSA 会调用 Msvl_0.dll 这个认证包,将用户信息加密处理,交由 SAM 服务器同 SAM 数据库中存储的密文进行对比。

(5) 如果对比后发现用户有效,SAM 会将用户的 SID(Security Identifier,安全标识符)、用户所属用户组的 SID 以及其他一些相关信息发送给 LSA。

(6) LSA 将收到的 SID 信息创建安全访问令牌,然后将令牌的句柄和登录信息发送给 Winlogon.exe。

(7) Winlogon.exe 对用户登录稍作处理后,完成整个登录过程。

2. 访问控制

Windows 系统的访问控制授权信息存储在 LSASS 的策略数据库中,客体的安全属性由安全控制项(ACE)来描述,全部客体的 ACE 组成访问控制表(ACL),没有 ACL 的客体意味着任何主体都可访问。内核中的安全引用监控器(Security Reference Monitor,SRM)根据 LSASS 的策略数据库,负责对所有安全主体访问 Windows 资源对象的访问进行控制。

3. 审计

Windows 系统的审计规则信息存储在 LSASS 的策略数据库中,内核中的安全引用监控器 SRM 根据 LSASS 服务配置的安全审计策略,对访问控制过程中相关事件进行记录,并由事件记录器(EventLog)生成系统审计日志。

6.8.2 Windows 系统安全机制

1. Windows 认证机制

早期 Windows 系统的认证机制不是很完善，甚至缺乏认证机制，例如 Windows 32、Windows 98 等。随着系统发展，微软公司逐步增强了 Windows 系统的认证机制。以 Windows XP 为例，系统提供两种基本认证类型：本地登录和基于活动目录的域登录。

1）本地登录

本地登录指用户登录的是本地计算机，对网络资源不具备访问权力。本地登录通常采用口令认证方式，所有系统合法用户的用户名与口令信息被存储在本地计算机的安全账户管理器（SAM）中，SAM 通常位于%SystemRoot%\system32\config 文件夹下，并在注册表的 HKEY_LOCAL_MACHINAE/SAM 中保存有副本。在登录时，用户提交登录凭证，本地计算机的安全子系统将用户名与口令送到本地计算机上的 SAM 数据库中进行验证。这里需要注意的是，Windows 的口令不是以纯文本格式存储在 SAM 数据库中的，而是以口令散列值的方式存储。同时为了保护 SAM，Windows 系统在运行时对 SAM 文件加了一个持久性的文件锁，因此即使是 Administrator 账户，通过正常途径也不能直接读取 SAM，只有 LocalSystem 账户权限才可以读取，但黑客们已经提出了多种技术可以从内存 dump 出 SAM 内容，从而使得对 SAM 文件进行暴力破解成为可能。

通过第 4 章身份认证的介绍，我们知道本地口令认证主要需解决弱口令问题，针对弱口令的攻击方式主要有字典攻击和暴力破解，为了防范这类攻击，Windows 系统提供了密码策略和账户锁定策略。

运行 secpol.msc 即可进行密码策略和账户锁定策略设置，本地安全设置中的密码策略是增强口令强度的策略，在默认的情况下都没有开启，需要开启的密码策略如表 6.4 所示。其中，“密码复杂性要求”要求设置的密码必须是大写字母、小写字母、特殊字符和数字的组合；“密码长度最小值”是密码长度至少要满足的要求；“密码最长留存期”决定了密码可以使用的最长时间（以天为单位），也就是说，当密码使用超过最长留存期后，就自动要求用户修改密码；“强制密码历史”是系统记录的历史密码数目，其目的是为了防止用户将几个密码轮换使用，表中“强制密码历史”值为 5，表示系统会记录 5 个最近使用的密码，用户设置新密码时不能与前 5 次密码相同。

表 6.4 Windows 密码策略

策　略	设　置
密码复杂性要求	启用
密码长度最小值	6 位
密码最长存留期	15 天
强制密码历史	5 个

开启账户策略可以在满足特定条件时将用户账户锁定，有效防止字典攻击和暴力破解，如表 6.5 所示。

表 6.5　Windows 账户策略

策　　略	设　　置
复位账户锁定计数器	30 分钟
账户锁定时间	30 分钟
账户锁定阈值	5 次

账户锁定阈值是指允许用户登录尝试的次数，如果登录失败次数超过阈值则该账户会被系统锁定，锁定时间由账户锁定时间策略决定。复位账户锁定计数器决定了需要等待多长时间，系统才自动将记录的失败次数清零，例如设置账户锁定阈值为 10 次，账户锁定时间为 60 分钟、复位账户锁定计数器为 30 分钟，如果一位用户忘记了自己的密码，尝试了 5 次就没有继续尝试，这时他的账户还没有没锁定，但系统已经记录了失败尝试的次数为 5，在最后一次尝试的 30 分钟后，记录下来的 5 次失败尝试会被清零，该账户又获得了 10 次尝试的机会。

如果操作系统的 SAM 数据库出现问题，将面临无法完成身份认证、无法登录操作系统、用户密码丢失的情况。所以，保护 SAM 数据的安全就显得尤为重要，虽然 SAM 数据库中用户口令已经做了散列处理，并且进行了运行时锁定，但是针对 SAM 进行破解的工具有很多，如 LophtCrack（LC）、Cain&Abel 等。Microsoft 公司提供了实用工具 SysKey 对 SAM 进行保护，读者可以自行完成 SysKey 的配置。

本地用户登录没有集中统一的安全认证机制。如果有 N 台计算机要相互访问资源，就需要在 N 台计算机上维护 N 个 SAM 库。这样，用户登录的验证机制就被分布到了多个地方，这违反了信息安全的可控性原则。因为在实际的环境中，如果需要在多个点上维护安全，还不如使用某一种机制在一个点上维护安全，以此达到统一验证、统一管理的目的。

2）基于活动目录的域登录

域是用来集中网络上资源与活动的计算机集合。域上可以有几台控制器和许多客户计算机。域控制器是指运行 Windows 服务器活动目录的机器。运行活动目录安全向导就能在计算机上安装和配置活动目录组件。域控制器存储资源目录、用户数据信息及域交互信息，包括用户登录过程、鉴别和目录检索。

活动目录是 Windows 服务器上运行的服务，存储域上大量对象的安全信息。它是个集中化的仓库，允许域控制器访问、存取和修改对象和它的属性。

基于活动目录的域登录与本地登录的方式完全不同。首先，所有的用户登录凭证（用户 ID 与口令）被集中地存放到一台服务器上，结束了分散验证的行为。该过程必须使用网络身份验证协议，这些协议包括 Kerberos、LAN 管理器（LM）、NTLAN 管理器（NTLM）等，而且这些过程对于用户而言是透明的。

如图 6.19 所示，此时网络上所有用户的登录凭证（包括用户 ID 和口令）都被集中地存储到活动目录安全数据库中。这台完成集中存储域用户 ID 与口令并提供用户身份的服务器，就是域控制器（DC）。用户在计算机上登录域时，需要通过网络身份认证协议，将登录凭证提交到 DC 进行认证。注意，在基于活动目录的域登录环境中必须部署活动目录服务器。

只要登录“域”成功，服务器之间或主机之间的相互访问就不再是进行分散的验证，而是通过活动目录去维护一个安全堡垒。如果计算机 B 与计算机 C 要相互访问活动目录上的

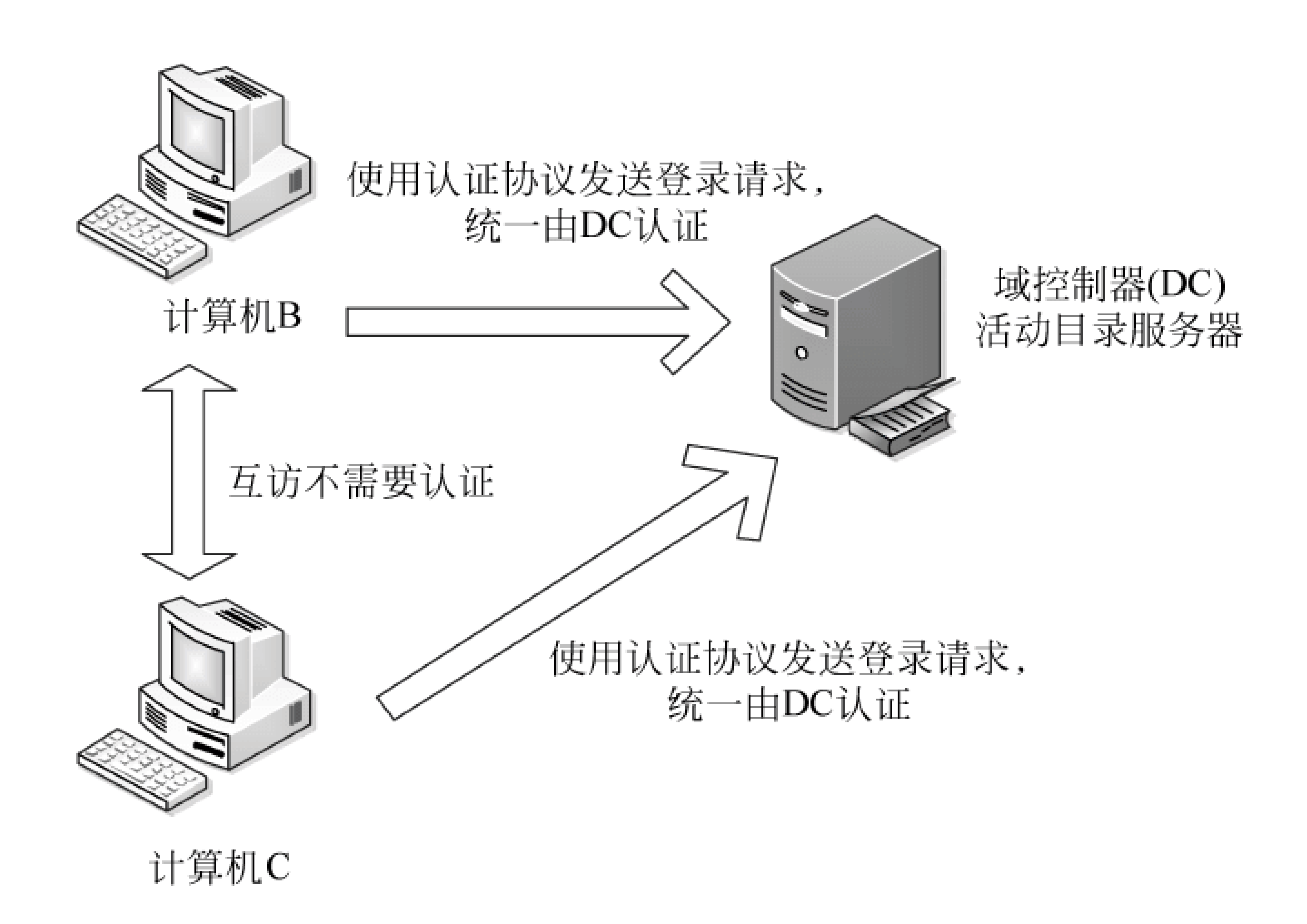


图 6.19 基于活动目录的域登录

资源，那么这两台主机在网络初始化时就必须成功地被域控制器所验证。那么此时计算机B与计算机C的相互访问，就不再需要输入用户名和密码了。这样就达到了“一次登录、多次访问”的效果，不仅提高了登录验证的安全性，也提高了访问效率。

2. Windows 访问控制机制

Windows 的访问控制策略是基于自主访问控制的，由属主用户决定其他用户是否可以访问资源以及对资源的访问能力，以保证资源合法、受控地使用。

Windows 安全子系统中实现自主访问控制主要包含 5 个关键的组件：安全标识符(SID)、访问令牌(Access Token)、安全描述符(Security Descriptor)、访问控制表(Access Control List)、访问控制项(Access Control Entry)。

1) 安全标识符(Security Identifiers，SID)

Windows 并不是根据账户名而是利用 SID 来区分账户的。在 Windows 环境下，几乎所有对象都具有唯一标识符 SID，例如本地账户、本地账户组、域账户、域账户组、本地计算机、域、域成员等。可以将用户名理解为每个人的名字，将 SID 理解为每个人的身份证号码，人名可以重复，但身份证号码绝对不会重复。这样做主要是为了便于管理，例如，因为 Windows 是通过 SID 区分对象的，完全可以在需要时更改一个账户的用户名，而不用再对新名称的同一个账户重新设置所需的权限，因为 SID 是不会变化的。然而，如果有一个账户，已经给该账户分配了相应的权限，一旦删除了该账户，然后重建一个使用同样用户名和密码的账户，原账户具有的权限和权力并不会自动应用给新账户，因为尽管账户的名称和密码都相同，但账户的 SID 已经发生了变化。

标志某个特定账号或组的 SID 是在创建该账号或组时由系统生成的。每个本地账号或组的 SID 在创建它的计算机上是唯一的，机器上的不同账号或组不能共享同一个 SID。SID 在整个生存期内也是唯一的。安全主体绝对不会重复发放同一个 SID，也不重用已删除账号的 SID。

SID 是一个形如“S-1-5-21-1416960467-3232414597-2644585607-500”的字符串，其中前缀 S 为固定字符，第二位为版本号，通常为 1，第三位表示颁发机构(5 代表 the Windows

Security Authority)，之后为 0 个或几个子颁发机构标识符，末尾是一个相对标识符(Relative Identifier，RID)，一般 Administrator 账户的 RID 均为 500。在 Windows 7 中，要查看当前登录账户的 SID，可以使用管理员身份启动命令提示窗口，然后运行“whoami/user”命令，如图 6.20 所示。

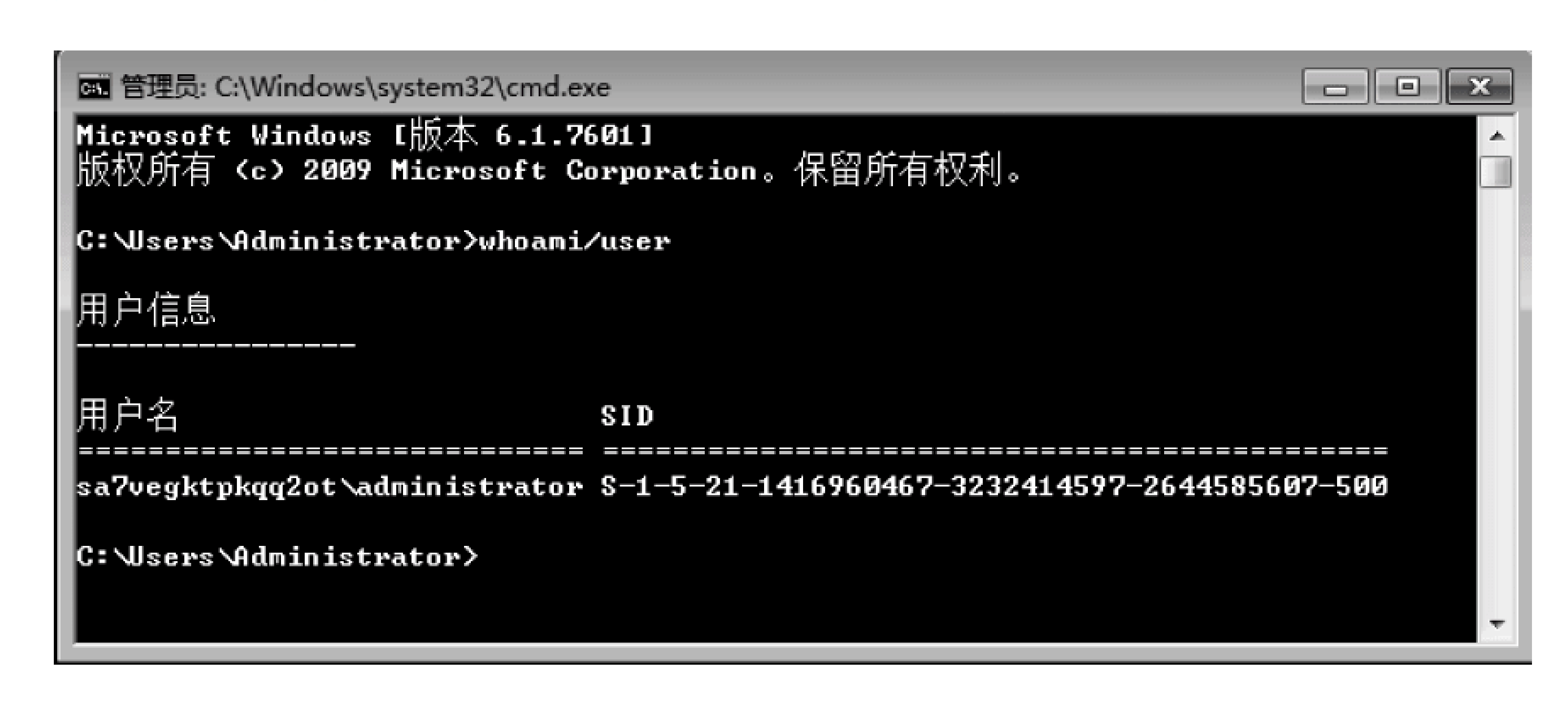


图 6.20　查看用户 SID

Windows XP 默认安装中没有 whoami 程序，因此如果想在 Windows XP 下查看当前账户的 SID，或者在 Windows Vista 和 Windows XP 下查看其他账户的 SID，可以借助微软的一个免费小工具 PsGetSid。

2) 访问令牌(Access Token)

安全引用监视器(SRM)使用一个称为访问令牌的对象来标识一个进程或线程的安全属性。访问令牌可以看作是一张电子通行证，里面记录了用于访问对象、执行程序、修改系统设置所需的安全验证信息。

所有的令牌包含了同样的信息，如图 6.21 所示，但是令牌的大小是不固定的，因为不同的用户账户有不同的权限集合，它们关联的组账户集合也不同。

用户的访问权限主要由令牌中的两部分信息来确定，第一部分由令牌的用户账户 SID 和组 SID 域构成。SRM 使用这些 SID 来决定一个进程或线程是否可以获得一个被保护对象(比如一个 NTFS 文件)的访问许可。令牌中的用户账户 SID 描述了用户身份，不同的用户访问权限不同，而组 SID 说明了一个用户的账户是哪些组的成员，用户组是为了简化用户管理而引入的用户账户容器，通过将用户账户添加到特定用户组，就可以使得该用户拥有用户组配置的全部权限，不同的组其权限也不相同，一个用户可以是多个组的成员，能够获得这些组的权限。

在一个令牌中，决定该令牌的线程或进程可以做哪些事情的第二部分信息是权限集。一个令牌的权限集是一组与该令牌关联的列表。关于权限的一个例子是，与该令牌关联的进程或线程具有关闭该计算机的权限。



图 6.21　访问令牌

令牌中包含的默认的主组域和默认的自主访问控制表(DACL)域是指这样一些安全属性：当该进程或线程创建对象时，Windows 系统自动将该进程或线程关联的令牌中的默认的主组和默认的自主访问控制表应用在它所创建的对象上，这样可以使得进程或线程可以很方便地创建一些具有标准安全属性的对象，而不需要为它所创建的每个对象请求单独的安全信息。

3）安全描述符(Security Descriptor)

令牌标识了一个用户的凭证，而安全描述符与一个对象关联在一起，规定了谁可以在这个对象上执行哪些操作。

一个安全描述符由以下属性构成，如图 6.22 所示。

版本号
标志
所有者 SID
组 SID
自主访问控制表(DACL)
系统访问控制表(SACL)

图 6.22 安全描述符

- 版本号：创建此描述符的 SRM 安全模型的版本。
- 标志：定义了该描述符的类型和内容。该标志指明是否存在 DACL 和 SACL。还包括如 SE_DACL_PROTECTED 的标志，防止该描述符从另一个对象继承安全设置。
- 所有者 SID：所有者的安全 ID，该对象的所有者可以在这个安全描述符上执行任何动作。所有者可以是一个单一的 SID，也可以是一组 SID。所有者具有改变 DACL 内容的权限。
- 组 SID：该对象的主组的安全 ID(仅用于 POSIX 系统)。
- 自主访问控制表(Discretinary ACL，DACL)：规定了谁可以用什么方式访问该对象。
- 系统访问控制表(System ACL，SACL)：规定了哪些用户的哪些操作应该被记录到安全审计日志中。

ACL 头
ACE 头
访问掩码
SID
ACE 头
访问掩码
SID
……

图 6.23 自主访问控制表

安全描述符的主要组件是自主访问控制表，自主访问控制表确定了各个用户和用户组对该对象的访问权限。当一个进程试图访问该对象时，以该进程的 SID 与该对象的自主访问控制表是否相匹配，来确定本次访问是否被允许。

4）自主访问控制表(Access Control List，ACL)

ACL 是 Windows 访问控制机制的核心，它的结构如图 6.23 所示。每个表由整个表的表头和许多访问控制项(ACE)组成。每个访问控制项为一个个人 SID 或组 SID 定义访问掩码，访问掩码定义了该 SID 被授予的权限。

当进程试图访问一个对象时，系统中该对象的管理程序从访问令牌中读取 SID 和组 SID，然后扫描该对象的 DACL，进行以下 3 种情况的判断。

- 如果目录对象没有访问控制表(DACL)，则系统允许所有进程访问该对象。
- 如果目录对象有访问控制表(DACL)，但访问控制条目 ACE 为空，则系统对所有进程都拒绝访问该对象。
- 如果目录对象有访问控制表(DACL)，且访问条目 ACE 不为空，那么如果找到一个访问控制项，它的 SID 与访问令牌中的一个 SID 匹配，那么该进程具有该访问控制项的访问掩码所确定的访问权限。

5）访问控制项(Access Control Entry,ACE)

访问控制项包含了用户或组的 SID 以及对象的权限。SID 用来标识允许、禁止或审计访问的用户或组。访问控制项有两种：允许访问和拒绝访问。拒绝访问的级别高于允许访问。

3. 用户账户管理(User Account Control)

由于历史原因,使用 Windows 的很多用户都直接以管理员权限运行系统,这对计算机安全构成很大隐患。从 Windows Vista 开始,Windows 加强了对用户账户控制的管理,使用“用户账户控制”(User Account Control,UAC)模块来管理和限制用户权限。

和老版本的 Windows 有很大不同,在 Windows Vista、Windows 7 等中,当用户使用管理员账户登录时,Windows 会为该账户创建两个访问令牌：一个标准令牌,一个管理员令牌。当用户试图访问文件或运行程序的时候,系统大都会自动使用标准令牌进行,只有在权限不足(也就是说,如果程序宣称需要管理员权限的时候)时,系统才会使用管理员令牌,这种将管理员权限区分对待的机制称为 UAC,UAC 体现了最小特权原则,即在执行任务时使用尽可能少的特权。

在需要管理员特权操作时,系统首先会弹出 UAC 对话框要求用户确认(如果当前登录的是管理员用户),如图 6.24 所示,或者输入管理员用户的密码(如果当前登录的是标准用户,也称为受限用户),只有在提供了正确的登录凭据后,系统才允许使用管理员令牌访问文件或运行程序,这个要求确认或者输入管理员账户密码的过程称为“提升”。

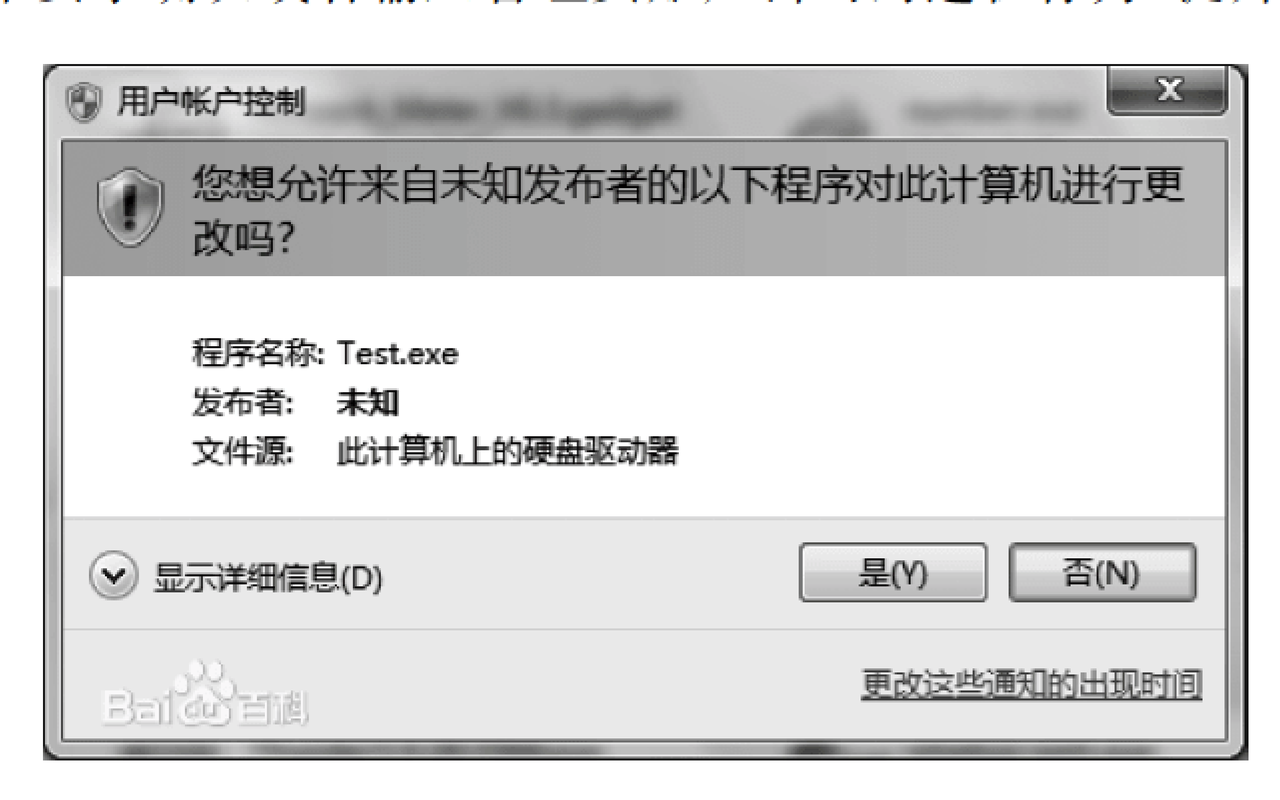


图 6.24 UAC 用户确认对话框

根据以管理员身份运行的程序不同,“提升”对话框顶部一栏的底色不同,一般来说,底色和对应的含义如表 6.6 所示。

表 6.6 UAC 对话框背景含义

背景颜色	含义
红色背景,带有红色盾牌图标	程序的发布者被禁止,或者被组策略禁止。遇到这种对话框的时候要万分小心
橘黄色背景,带有红色盾牌图标	程序不被本地计算机信任(主要是因为不包含可信任的数字签名或数字签名损坏)
蓝绿色背景	程序是微软自带的,带有微软的数字签名
灰色背景	程序带有可信任的数字签名

UAC功能在一定程度上增强了系统安全性，但是在执行很多操作的时候都需要进行确认，也带来了使用上的麻烦。在Windows 7系统中可以根据系统所处的环境状况，设置UAC的安全级别，使得在利用UAC确保系统安全的同时，使用上更加易于接受。使用管理员账户登录Windows 7，打开【控制面板】，在【控制面板】中依次单击【用户账户】、【更改用户账户控制设置】，在如图6.25所示的界面中，通过滑块调整UAC的提示级别，系统中提供了4个级别，从上到下的安全性递减，同时，“扰民”的程度也是递减的。

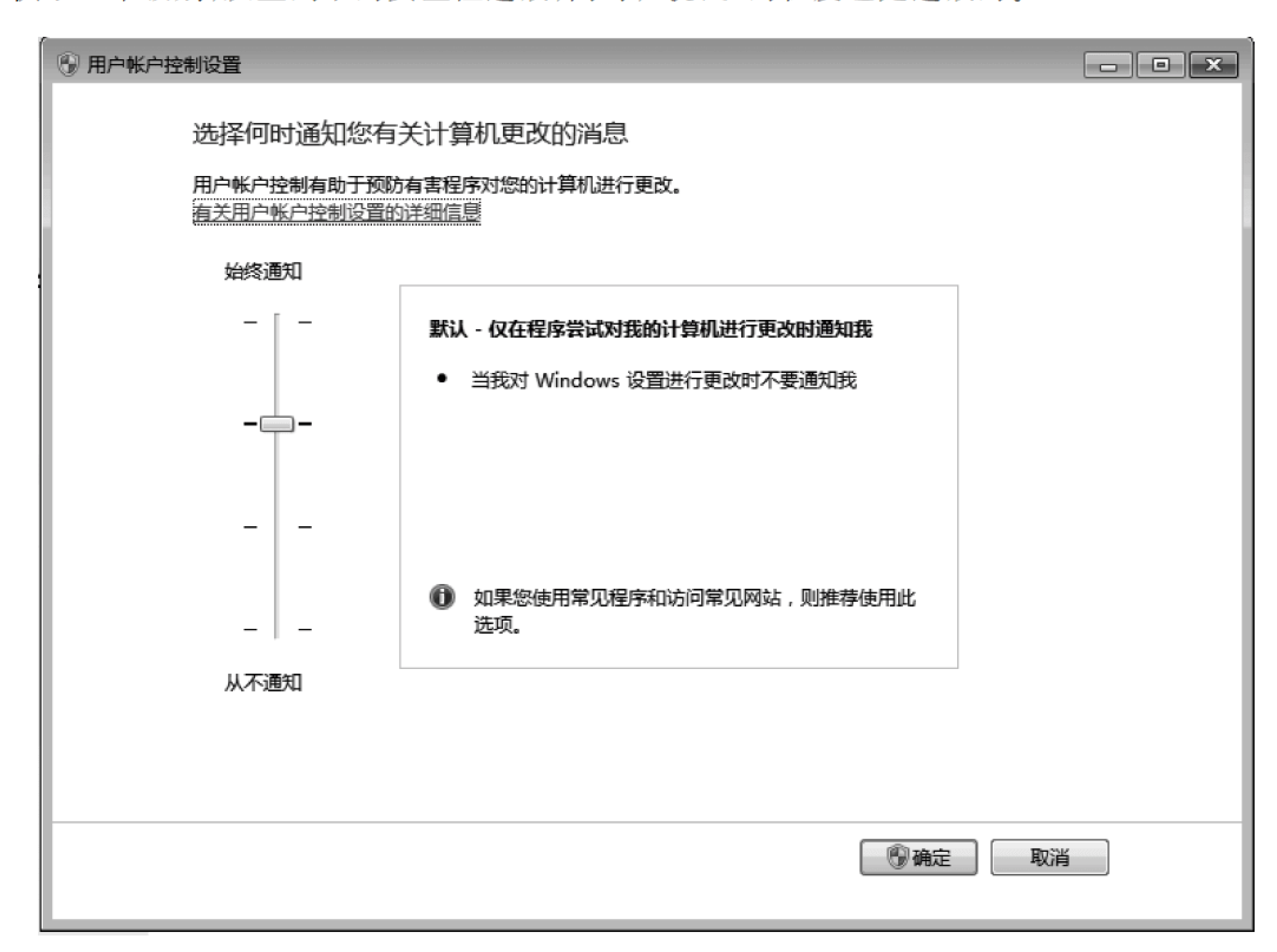


图6.25 UAC级别设置

不同级别之间的区别以及建议的使用环境可参考表6.7。

表6.7 UAC各级别使用场景

选项	描述	适用场景	是否使用安全桌面
始终通知	当程序试图安装软件，或更改计算机设置，或用户更改Windows设置时，通知当前用户	如果希望尽可能保证计算机安全，用户需要频繁安装软件，以及访问不熟悉的网站时，可使用该选项	是
默认值	只有在程序试图修改计算机配置时通知当前用户，但用户自己更改Windows设置时不通知	如果计算机需要较高的安全等级，并希望降低用户可以看到的通知数量时，可选择该选项	否

续表

选　　项	描　　述	适 用 场 景	是否使用安全桌面
仅当程序尝试更改计算机时通知(不降低桌面亮度)	与默认值相同,但显示通知时 UAC 不切换到安全桌面	如果用户在可信赖环境中工作,只使用熟悉的应用程序,不访问不熟悉的网站,即可选择该选项	否
从不通知	关闭 UAC 所有的提示通知	如果安全性并不是最重要的,并且用户在可信赖环境中工作,同时使用由于不支持 UAC 而无法获得 Windows 7 认证的程序,即可使用该选项	否

Windows 7 默认 UAC 级别是第三级,在该级别下,当弹出 UAC 提升对话框时,桌面背景会变暗,这就是所谓的"安全桌面",这样做的主要目的不是为了突出显示 UAC 的对话框,而是为了安全,除了受信任的系统进程外,任何用户级别的进程都无法在安全桌面上运行,这样可以阻止恶意程序的仿冒攻击。

4. 加密文件系统

权限的访问控制行为在某种程度上提高了资源的安全性,防止了资源被非法访问或修改。但事实上这种行为只能针对系统级层面进行控制,如果资源所在的物理硬盘被非法窃取,那么使用上述的权限访问控制行为对资源保护没有任何意义。窃取者只需把资源所在的物理硬盘放到自己的主机上,使用自己的操作系统启动计算机,再将该硬盘设置成为操作系统资源盘,便可以轻松地解除原有操作系统的权限并访问资源。为了防止这样的事情发生,需要一种基于文件系统加密的方法来保证资源的安全。

加密文件系统(Encrypting File System,EFS)是 Windows 2000 及以上版本中 NTFS 格式磁盘的文件加密。EFS 允许用户以加密格式存储磁盘上的数据,将数据转换成不能被其他用户读取的格式。用户加密文件之后只要文件存储在磁盘上,它就会自动保持加密的状态。

EFS 基于混合加密体制,文件加密密钥为 124 位,加密算法是 DES 的改进算法,文件加密密钥用公钥密码体制的公钥加密后存储。注意,由于加密、解密功能在系统启动时还不起作用,因此系统文件或在系统目录中的文件是不能被加密的,否则,系统将无法启动。

加密时,只需使用鼠标右键单击要加密的文件或文件夹,然后选择【属性】,在【属性】对话框的【常规】选项卡中单击【高级】按钮,在【高级属性】对话框中选中【加密内容以保护数据】复选框并确认即可对文件进行加密,如图 6.26 所示,如果加密的是文件夹,系统将进一步弹出【确认属性修改】对话框要求确认是加密选中的文件夹,还是加密选中的子文件夹以及其中的文件。解密的步骤与加密相反,只需在【高级属性】对话框清除【加密内容以保护数据】复选框上的选中标记即可。在解密文件夹时将同样弹出【确认属性更改】对话框要求确认解密操作应用的范围。

5. BitLocker 机制

为了保证系统安全,可以采用给用户设置强密码、访问控制、EFS 加密等手段,然而这样就可以做到万无一失了吗?很多人可能听说过 ERD Commander 之类的软件,这种软件

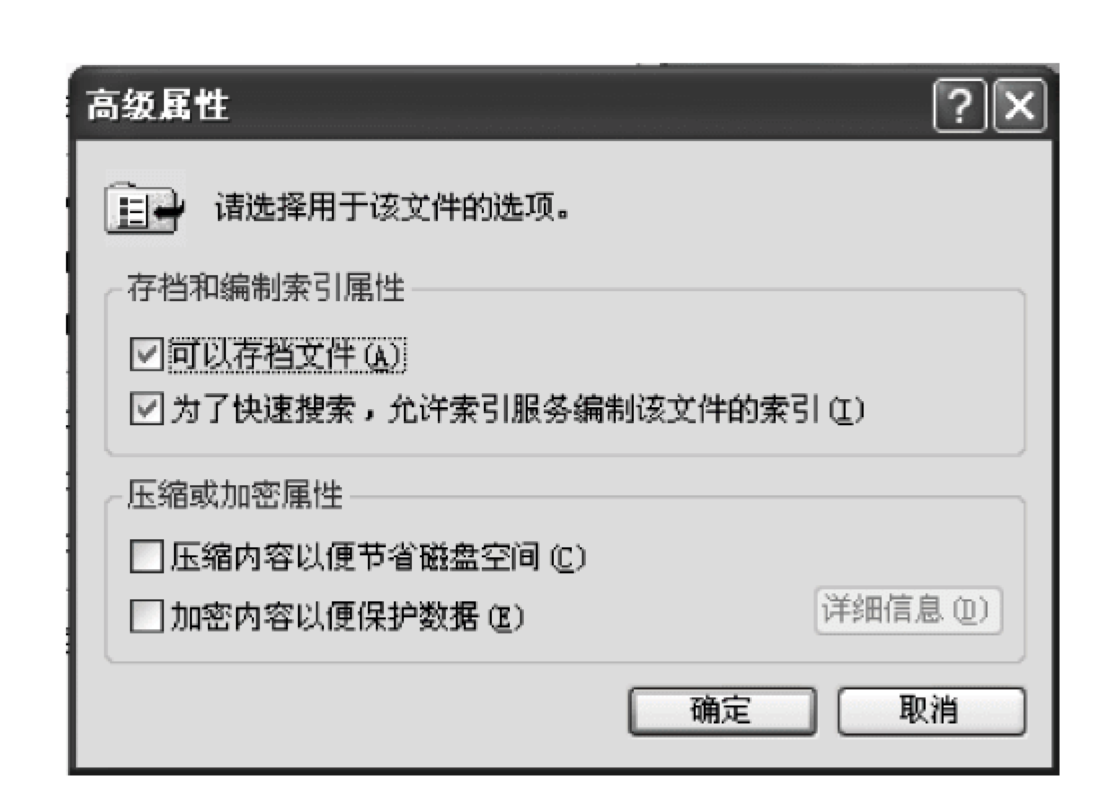


图 6.26 加密文件系统

可以创建一个光盘镜像文件，将其刻录到光盘上后，可以用来引导计算机启动，将计算机引导进入一种 Windows PE 环境(可以理解为运行在光盘上的 Windows 系统)，利用该环境，可以在硬盘上原先安装的 Windows 没有启动的情况下，查看注册表内容、访问文件，查看 EFS 加密密钥等，这种攻击是在硬盘 Windows 没有运行的情况下进行的，因此，称为“脱机攻击”。

Windows 7 中新增的 BitLocker 功能可以加密整个 Windows 分区，并将加密密钥保存在硬盘之外的地方，这样即使遭受脱机攻击，攻击者没有解密密钥，也无法获得系统中的任何信息。

BitLocker 主要有两种工作模式：TPM 模式和 U 盘模式。要使用 TPM 模式，要求计算机中必须带有不低于 1.2 版 TPM 芯片，这种芯片是通过硬件方式提供的，一般只出现在对安全性要求较高的商用计算机或者工作站上，家用计算机或者普通计算机上通常不会提供。要想知道计算机上是否有 TPM 芯片，需要运行 devmgmt.msc 打开设备管理器，然后查看设备管理器中是否存在“安全设备”节点，该节点下是否有“受信任的平台模块”这类设备，并确定其版本即可。如果要使用 U 盘模式，只需要计算机上有 USB 接口，计算机的 BIOS 支持在开机的时候访问 USB 设备，并且能提供一个专用的 U 盘。使用 U 盘模式，用于解密系统盘的密钥文件会被保存在 U 盘上，每次重启系统的时候，必须在开机之前将 U 盘连接到计算机上。

打开【控制面板】，单击【BitLocker 驱动器加密】，即可启动磁盘加密过程。在应用了 U 盘模式的 BitLocker 后，每次启动系统前都必须将保存了启动密钥的 U 盘连接到计算机，才能完成 Windows 的启动和加载过程。

6. Windows 审计/日志机制

日志文件是 Windows 系统中一个比较特殊的文件，它记录 Windows 系统的运行状况，如各种系统服务的启动、运行和关闭等信息。Windows 日志有 3 种类型：系统日志、应用程序日志和安全日志。可以通过“控制面板”→“管理工具”→“事件查看器”来浏览这些日志文件中的内容。

(1) 系统日志。包含 Windows 系统组件记录的事件。例如，在启动过程中加载驱动程序或其他系统组件失败将记录在系统日志中，默认情况下 Windows 会将系统事件记录到系统日志中。

(2) 应用程序日志。包含由应用程序或系统程序记录的事件，主要记录程序运行方面

的事件。

(3) 安全性日志。记录诸如有效和无效的登录尝试以及与资源使用相关的事件,例如创建、打开或删除文件或其他对象。

安全审核是 Windows 最基本的入侵检测方法,当有人尝试对系统进行某种方式(如尝试用户密码、改变账户策略和未经许可的文件访问等)入侵时,都会被安全审核记录下来。审核策略在默认的情况下是没有开启的,可利用本地安全策略中的审核策略开启,如图 6.27 所示。



图 6.27　审核策略

如果计算机被配置为域控制器,那么还将包括目录服务日志、文件复制服务日志。如果计算机被配置为域名系统(DNS)服务器,那么还将记录 DNS 服务器日志。当启动 Windows 时,事件日志服务(EventLog)会自动启动,所有用户都可以查看应用程序和系统日志,但只有管理员才能访问安全性日志。

7. Windows 协议过滤和防火墙

针对来自网络上的威胁,Windows NT 4.0、Windows 2000 提供了包过滤机制,通过过滤机制可以限制网络包进入到用户计算机。而 Windows XP SP2 以后的版本则自带了防火墙,该防火墙能够监控和限制用户计算机的网络通信。

6.9　UNIX/Linux 的安全机制

UNIX 是一种多用户、多任务的操作系统,这类操作系统的一个基本功能就是防止系统中不同用户之间的相互干扰,所以 UNIX 的设计宗旨是要考虑安全的。

Linux 是 UNIX 的一个克隆版本,在安全结构上,Linux 与 UNIX 基本相似,如无特殊说明,下面对 UNIX 安全机制的描述同样也适用于 Linux。

6.9.1　UNIX 与 Linux 操作系统概述

1965 年,美国 AT&T 贝尔实验室(Bell Labs)、通用电气公司(General Eletric)和麻省理工学院(MIT)等合作开发名为 MULTICS 的用于大型机的操作系统,MULTICS 的主要

设计目标是提供一种用于多用户多任务的大规模计算环境，方便实现硬、软件资源的共享。由于项目过于庞大，难以达到预期目标，1969 年，因 MULTICS 计划的工作进度太慢，该计划被停了下来。MULTICS 虽然失败了，但是给程序员积累了宝贵的经验。

1969 年，Ken Thompson、Dennis Ritchie 和其他参与 MULTICS 计划的一些人重回贝尔实验室，在一台闲置无用的 DEC PDP-7 机器上研制“太空旅行”游戏，为了改善研制环境，他们经过一番艰苦的努力，给这台机器开发了新的操作系统，这就是最初的 UNIX 系统，从 UNIX 的名字就可以知道，这个小型操作系统是从对过于庞大的 MULTICS 的反省中诞生的（UNIX 表示“单一”，而 MULTI 表示众多），所以它的功能被大幅度简化，成为非常单纯的 OS。

1971 年，Dennis Ritchie 开发了 C 语言，并于 1973 年用 C 语言重写了 UNIX，从而 UNIX 与 C 语言就紧密地结合在一起，这一实现也是 UNIX 变成开放系统的重要原因。

由于一开始无法进入计算机市场，AT&T 无法将 UNIX 作为商品出售，于是贝尔实验室开始向大学、科研机构免费发放 UNIX 的源代码，这使得大批优秀的计算机人员在 UNIX 系统上被培养出来，从而为今后更广泛地应用、开发 UNIX 系统打下良好的基础。

1975 年，应学术界要求，贝尔实验室推出了 UNIX Version 6，直到 1977 年 UNIX 才得到商业许可，1979 年为满足商业需求推出了 UNIX Version 7。在 20 世纪 80 年代初，AT&T 开发了 UNIX 的后续版本 UNIX System Ⅲ和 System Ⅴ。在 20 世纪 80 年代末，AT&T 对 System Ⅴ的命名重新标准化，以 System Ⅴ Release Ⅹ的形式表示，简记为 SVRX，这些版本都是以第 7 版为基础发展而成的，System Ⅴ3.2 和 System 4.2 在计算机操作系统中一直很流行。

在 AT&T 发展 UNIX 的同时，许多大学也在研究 UNIX，加利福尼亚大学的程序员改动 AT&T 发布的源代码，开发了 UNIX 的伯克利发布版（Berkeley Standard Distribution，BSD），成为第二个主要的 UNIX 版本。

在 UNIX 的发展历史中，还产生了许多其他的商业版本，如 Sun Microsystems 公司的 SunOS/Solaris、IBM 公司的 AIX、SCO 公司的 SCO OpenServer 及 UNIXWare 7 等。

Linux 是类 UNIX 系统，或者说它是 UNIX 操作系统的一个克隆版本。在 Linux 诞生之前，为了教学和研究的需要，阿姆斯特丹 Vrije 大学的计算机科学家 Andrew S. Tanwnbaun 以 UNIX 为蓝本开发了 Minix 作为教育工具。1991 年初，芬兰赫尔辛基大学学生 Linus 开始在一台 386sx 兼容微机上学习 Minix 操作系统。通过学习，他逐渐不能满足于 Minix 系统的现有性能，并开始酝造开发一个新的免费操作系统，1991 年，Linus 在 Minix 新闻组上发帖，要求试用他写的操作系统，得到了计算机爱好者和黑客的热烈响应，从此，改变了整个计算科学领域。

1993 年 Linux 的第一个“产品”版 Linux 1.0 问世，在这个版本发布之初，系统是按完全自由扩散版权进行扩散的，它要求所有的源码必须公开，而且任何人都不能从 Linux 交易中获利，但很快 Linux 的创始人 Linus 开始意识到这种纯粹的自由软件发布方式对于 Linux 的扩散和发展大大不利，因为它使 Linux 无法以磁盘拷贝或者 CD-ROM 等媒体形式进行扩散，同时也因无利益驱动使一些商业公司敬而远之。于是 Linus 决定转向 GPL 版权，这一版权除了规定有自由软件的各项许可之外，还允许用户出售自己的程序副本。事实证明，这一版权上的转变对于 Linux 的进一步发展而言确实极为重要，从此以后，便有多家技术力量雄厚又善于市场运作的商业软件公司加入了原先完全由业余爱好者和网络黑客组成的

Linux 开发集团。紧接着多种 Linux 版本如雨后春笋一般出现在市场上，这些版本增加了更易于用户使用的图形界面和众多的软件开发工具，极大地拓展了 Linux 的功能和影响。

世界上成千上万的人在为 Linux 开发软件，如果每个软件都让 Linux 使用者自己获取安装，显然是不现实的。于是有了专门的公司或 Linux 爱好者，在 Linux 内核的基础上，加入 Linux 安装程序，选配了 Linux 下开发的大量应用软件包，以及方便的管理工具等，并提供一定的技术支持，久而久之，形成了不同的发行版本，RedHat、Debian 等都是比较著名的 Linux 版本。

6.9.2　UNIX/Linux 安全机制

1. 标识和鉴别

UNIX 中拥有最高权限的是超级用户（root），其功能和 Windows NT 的管理员（Administrator）功能类似，作为超级用户可以执行任何操作，管理一切资源，包括用户账号、文件和目录、网络资源等。超级用户一般在安装系统时创建，其他用户为普通用户，通常是系统安装完成后由管理员创建，这类用户只能访问和管理有限资源，也称受限用户。

在创建普通用户时需要设置用户名和口令信息，在系统内部具体实现中，系统会为每个用户分配一个唯一的标识号-UID，例如超级用户（root）的 UID 为 0。每个用户可以属于一个或多个用户组，每个组由 GID 唯一标识。用户号（UID）和用户组号（GID）决定了用户的访问权限。

所有与用户相关的信息存储在系统的/etc/passwd 文件中，包含用户的登录名、经过加密的口令等。这个文件的拥有者是超级用户，只有超级用户拥有写的权力，而普通用户只有读的权力。

```
#ls -l /etc/passwd
# -rw-r--r-- root root
```

那么，/etc/passwd 文件中具体包含哪些内容呢？

```
#vi /etc/passwd
```

如图 6.28 所示，该文件是一个典型的数据库文件，每一行都由 7 个部分组成，每两部分之间用冒号分隔开，这 7 个部分分别描述了以下信息：用户名、口令、用户 ID、用户组 ID、用户描述、用户主目录、用户的登录 Shell，下面分别描述。

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
```

图 6.28　/etc/passwd 文件内容

(1) 用户名：用户的登录名。

(2) 口令：用户的口令，以加密形式存放。该域值如为 x 表示口令存储在/etc/shadow 中。

(3) 用户 ID(UID)：系统内部以 UID 标识用户，范围为 0～32 767 之间的整数。

(4) 用户组 ID(GID)：标志用户所在组的编号。将用户分组管理是 UNIX 系统对权限管理的一种有效方式。假设有一类用户都要赋予某个相同的权限，如果给用户分别处理，将会很复杂，如果把这些用户都放入一个组中，再给组授权，就容易多了。一个用户可以属于多个不同的组。组的名称和信息放在另一个系统文件/etc/group 中，与用户标识符一样，GID 的范围也是 0～32 767 之间的整数。

(5) 用户描述：这个域中记录的是用户本人的一些情况，如用户名称、电话和地址等。该域的作用随着系统功能的增强，已经失去了原来的意义。一般情况下，约定该域存放用户的基本信息，也有的系统不需要该域。

(6) 用户主目录：这个域用来指定用户的主目录(home)，当用户成功进入系统后，他就会处于自己的用户主目录下。

一般情况下，管理员将在一个特定的目录里依次建立各个用户的主目录，目录名一般就是用户的登录名。用户对自己的主目录有完全控制的权限，其他用户对该目录的权限需要管理员手动分配。

如果没有指定用户的主目录，用户登录时将可能被系统拒绝或获得对根目录的访问权，这是非常危险的。

(7) 用户的登录 Shell：Shell 程序是一个命令行解释器，它能够读取用户输入命令，并将执行结果返回给用户，实现用户与操作系统的交互，它是用户进程的父进程，用户进程多由 Shell 程序来调用执行。在 UNIX 系统中有很多 Shell 程序，如/bin/sh、/bin/csh、/bin/ksh 等，每种 Shell 程序都具有不同的特点，但基本功能是一样的。

在用户登录时，输入用户名、口令信息，用户名是标识，它告诉计算机该用户是谁，而口令是确认数据。当用户输入口令时，UNIX 使用改进的 DES 算法(通过调用 crypt()函数实现)对其加密，并将结果与存储在数据库中的加密用户口令进行比较，若两者匹配，则说明该用户为合法用户，否则为非法用户。

为了防止口令被非授权用户盗用，对其设置应以复杂、不可猜测为标准。一个好的口令应该满足长度和复杂度要求，并且定期更换，通常，口令以加密的形式表示，由于/etc/passwd 对任何用户可读，故常成为口令攻击的目标，在后期的 UNIX 版本以及所有 Linux 版本中，引入了影子文件的概念，将密码单独存放在/etc/shadow 中，而原来/etc/passwd 文件中存放口令的域用 x 来标记。文件/etc/shadow 只对 root 用户开放读权，对普通用户不可读，以进一步增强口令的安全。

```
#ls -l /etc/shadow
#-rw-r----- root root
```

/etc/shadow 每一行记录包含 9 个字段，用分号隔开，如图 6.29 所示，分别描述用户名、加密后的口令信息、口令的有效期等信息。

在 Linux 系统中可以设置用户锁定策略、自动注销时间等以增强身份认证的安全性。

```
root:$1$rsvQv1rO$UqVal6mm1ckLxQIzzHQWa0:12524:0:99999:7:::
bin:*:12524:0:99999:7:::
daemon:*:12524:0:99999:7:::
adm:*:12524:0:99999:7:::
lp:*:12524:0:99999:7:::
sync:*:12524:0:99999:7:::
shutdown:*:12524:0:99999:7:::
halt:*:12524:0:99999:7:::
mail:*:12524:0:99999:7:::
news:*:12524:0:99999:7:::
uucp:*:12524:0:99999:7:::
operator:*:12524:0:99999:7:::
games:*:12524:0:99999:7:::
gopher:*:12524:0:99999:7:::
ftp:*:12524:0:99999:7:::
nobody:*:12524:0:99999:7:::
```

图 6.29　/etc/shadow

1）设置账户登录失败锁定、锁定时间

编辑文件/etc/pam.d/system-auth，查看有无“auth required pam-tally.so”条目的设置，如无则添加该条目并设置为需要的策略如：

＃vi /etc/pam.d/system-auth

设置：auth required pam-tally.so onerr＝fail deny＝6 unlock-time＝300

将设置密码连续错误 6 次锁定，锁定时间为 300 秒。

2）设置自动注销时间

如果用户在离开系统前忘记注销账户，会带来很大的安全隐患，应该设置成系统能够自动注销账户。

编辑文件/etc/profice，查看有无 TIMEOUT 条目的设置，如无则添加该条目并设置成需要的策略。

＃vi /etc/profice

并设置：TMOUT＝600

则系统中登录用户在 600 秒内没有任何操作，则系统自动注销该用户。

2. 访问控制

UNIX 系统的资源访问请求是基于文件的，在 UNIX 系统中各种硬件设备、端口甚至内存都是以设备文件的形式存在的，虽然这些文件和普通文件在实现上是不同的，但它们对外提供的界面是一样的，这样就给 UNIX 系统资源的访问控制带来了实现上的方便。

UNIX/Linux 提供的访问控制机制为自主访问控制，采用访问控制列表方式实现，将用户进行分组授权，一般分为属主用户、同组用户和其他用户三类，这种访问控制的粒度比较粗，无法实现对单个用户的授权。

1）访问权限

命令 ls 可列出文件(或目录)对系统内不同用户所给予的访问权限，如：

-rw-r--r--1　root　root　1397　Mar　7 10:20　passwd

图 6.30 给出了文件访问权限的图示解释。

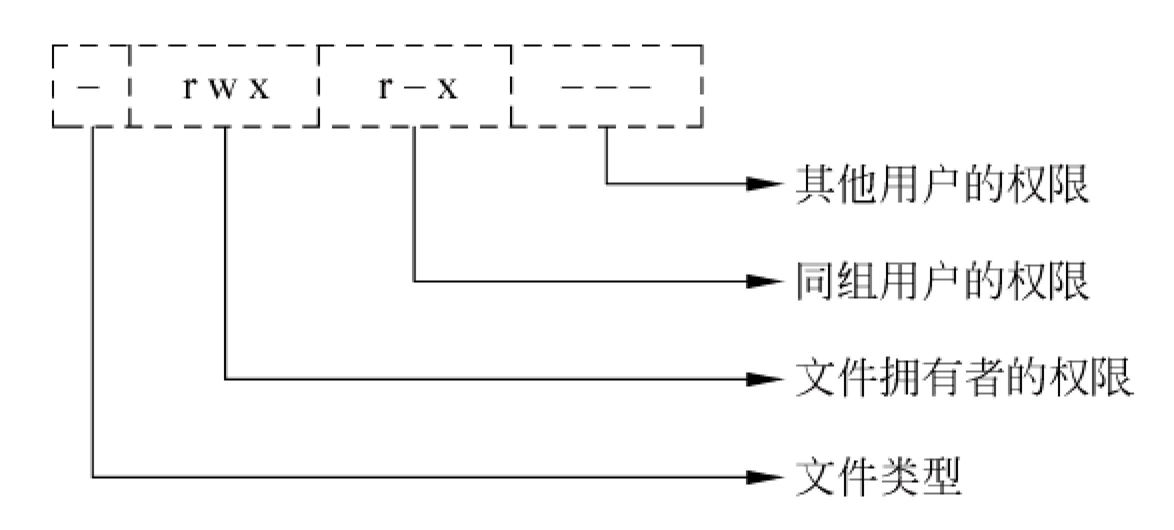


图 6.30 Linux 自主访问控制

访问权限位共有 9 位,分为三组,用以指出不同类型的用户对该文件的访问权限。

权限有三种:

(1) r 允许读;

(2) w 允许写;

(3) x 允许执行。

用户有三种类型:

(1) owner:该文件的属主;

(2) group:与该文件属主同组的用户,即同组用户;

(3) other:除以上两者外的其他用户。

图 6.30 表示文件的属主具有读写及执行权限(rwx),同组用户允许读和执行操作(rx),其他用户没有任何权限。在权限位中,-表示相应的访问权限位不允许。

上述授权模式同样适用于目录,目录的文件类型为 d。对于目录的读权限是指用 ls 列出目录中的内容,在目录中增删文件需要有写权限。进入目录或将该目录作路径分量时要求有执行许可,因此要使用任一文件,必须有该文件及找到该文件所在路径上所有目录分量的相应许可。仅当要打开一个文件时,文件的许可才开始起作用,而 rm、mv 只要有目录的搜索和写许可,并不需要有关文件的许可,这一点应尤其注意。

超级用户对任何文件或目录均可进行任何操作,具有最高权力,这样方便了管理员对系统的管理,但同时也是一个潜在的安全隐患。对于 root 账户的使用,需要注意以下几点:

(1) 除非必要,尽量避免以 root 用户身份登录;

(2) 不要随意将 root Shell 留在终端;

(3) 不要以 root 身份运行其他用户的或不熟悉的程序。

2) 改变权限

改变文件的访问权限可使用 chmod 命令,并以新权限和该文件名为参数,格式为:

```
chmod  [-Rfh]  访问权限  文件名
```

chmod 也有其他方式的参数可直接对某组参数进行修改,详见 UNIX 系统的联机手册。合理的文件授权可防止偶然性的覆盖或删除文件,改变文件的属主和组名可用 chown 和 chgrp,但修改后原属主和组员就无法修改回来了。

文件的授权可用一个 4 位的八进制数表示,后三位同图 6.30 所示的三组权限,授以权

限时许可位置 1，不授以权限则相应位置 0。最高的一个八进制数分别对应 SUID 位、SGID 位、sticky 位，其中前两个与安全有关，将其作为特殊权限位在下一节中进行描述。

umask（UNIX 对用户文件模式屏蔽字的缩写）也是一个 4 位的八进制数，UNIX 用它确定一个新建文件的授权。每一个进程都有一个从它的父进程中继承的 umask。umask 说明要对新建文件或新建目录的默认授权加以屏蔽的部分。

新建文件的真正访问权限＝(～umask)&(文件授权)

UNIX 中相应有 umask 命令，若将此命令放入用户的.profile 文件，就可控制该用户后续所建文件的访问许可。umask 命令与 chmod 命令的作用正好相反，它告诉系统在创建文件时不给予什么访问权限。

3）特殊权限位

有时没有被授权的用户需要完成某些要求授权的任务，如对于普通用户，应该允许通过 passwd 命令改变自身的口令，但是这势必涉及到对/etc/passwd 文件的修改操作，而普通用户不拥有直接修改/etc/passwd 文件的权力，以防止改变其他用户的口令。为了解决这个问题，UNIX 允许对可执行的目标文件（只有可执行文件才有意义）设置 SUID（Set User ID）或 SGID（Set Group ID），允许用户以特殊权限来运行程序。

如前所述，当一个进程执行时就被赋予 4 个编号，以标识该进程隶属于谁，分别为实际和有效的 UID、实际和有效的 GID。有效的 UID 和 GID 一般和实际的 UID 和 GID 相同，有效的 UID 和 GID 用于系统确定该进程对于文件的访问许可。而设置可执行文件的 SUID 许可将改变上述情况，当设置 SUID 时，进程的有效 UID 为该可执行文件的所有者的有效 UID，而不是执行该程序的用户的有效 UID，这样程序的所有者将可通过程序的控制在有限的范围内向用户发布不允许被公众访问的信息。同样，SGID 是设置有效的 GID。

```
#ls -a /usr/bin/passwd
#-rwsr-xr-x 1 root root
```

/usr/bin/passwd 就是一个 s 位程序，普通用户执行该程序时，进程的有效 UID 就改变为/usr/bin/passwd 这个可执行文件的属主 root，因此可以修改自己的口令。

用“chmod u+s 文件名”和“chmod u−s 文件名”来设置和取消 SUID 设置，用“chmod g+s 文件名”和“chmod g−s 文件名”来设置和取消 SGID 设置。

SUID 程序会使普通用户权限得到提升，从而给系统安全带来威胁，为了保证 SUID 程序的安全性，系统管理员应对系统中所有设置 SUID 或 SGID 的程序进行定期的检查和监视，严格限制功能范围，不能有违反安全性规则的 SUID 程序存在，并且要保证 SUID 程序自身不能被任意修改。

3. 审计

审计是 UNIX 安全机制的重要组成部分，它通过对安全相关事件进行记录和分析，发现违反安全策略的活动，确保安全机制正确工作并能对系统异常及时报警提示。审计记录常写在系统的日志文件中，丰富的日志为 UNIX 的安全运行提供了保障。常见的日志文件如表 6.8 所示。

表 6.8 审计文件

日志文件	说明
acct 或 pacct	记录每个用户使用过的命令
aculog	筛选出 modems(自动呼叫部件)记录
lastlog	记录用户最后一次成功登录和最后一次登录失败的时间
loginlog	记录不良的登录尝试记录
messages	记录输出到系统主控台及由 syslog 系统服务程序产生的信息
sulog	记录 su 命令的使用情况
utmp	记录当前登录的每个用户
wtmp	记录每一次用户登录和注销的历史信息,以及系统关和开
xferlog	记录 ftp 的访问情况

Linux 系统中传统的审计机制是 Syslogd 和 Klogd。Syslog 是一个应用层的审计机制,允许应用程序将审计信息传递给系统日志守护程序 Syslogd,由 Syslogd 根据配置文件(/etc/syslogd.conf)将收到的信息按类型作相应处理,写入不同的日志文件,如图 6.31 所示。另外,也允许内核消息守护进程 Klogd 将内核中通过 Printk 打印出的消息写入日志文件。

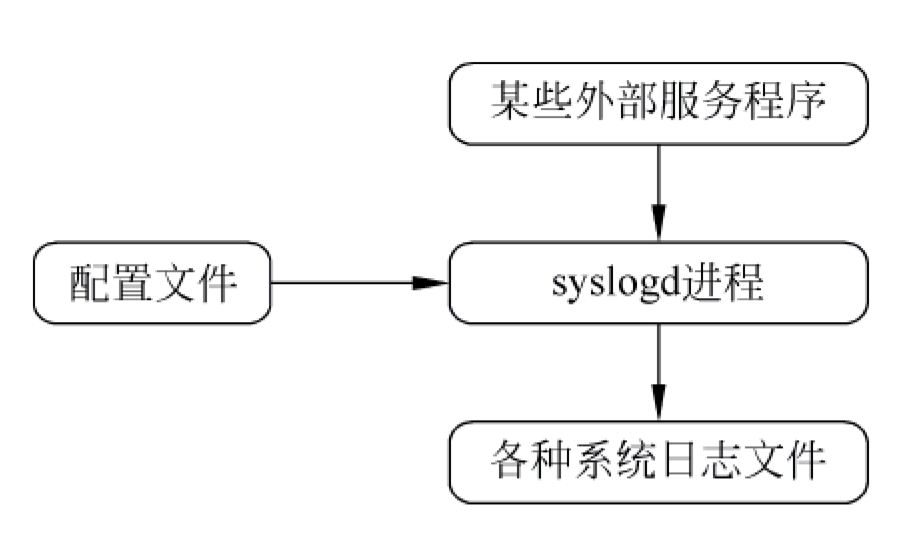


图 6.31 审计机制

Linux 传统的审计方式具有很大的局限性。首先,它不能提供系统级审计记录。Syslogd 只能接收由应用程序产生的日志信息,而 Klogd 只能接收内核中由 Printk 打印出的消息,这两种方式审计信息量获取有限,审计记录不够详细。其次,就应用层审计而言,Syslog 也是有局限性的。Syslog 产生的审计信息完全依赖于应用程序。如果入侵者熟悉 Syslog 的工作方式,就可以模仿与某个应用程序相同的方式写入日志,伪造出虚假的审计数据。一旦某个收集审计数据的外部服务程序被恶意用户杀掉后,由该服务程序所收集的某类审计记录就不会产生,这样审计系统也就达不到记录所有安全相关系统活动的目的。

为了达到 TCSEC 所规定的 C2 级的审计标准,当前的 UNIX/Linux 系统都对传统审计机制进行了改进和增强。

6.10 隐蔽信道

信息系统使用自主访问控制和强制访问控制策略来约束合法通道中的信息流动,合法通道主要是指文件和共享内存等。人们在实践中发现,恶意用户还可以利用信息系统中原本不用于通信的通道来传递信息,例如系统存储位置、定时设备等,常称为隐蔽信道。

隐蔽信道的概念最早是由 Lampson 提出的，他将隐蔽信定义为：如果一个通信信道不是被设计用来传输信息的，那么此信道是隐蔽的。1985 年美国国防部发布的可信计算机系统安全评价标准（TCSEC）中将隐蔽信道定义为：任何在违背系统安全策略的情况下被用来传输信息的通信信道。我国的计算机信息系统安全保护等级划分准则（GB 17859-1999）中把隐蔽信道定义为：允许进程以危害系统安全策略的方式传输信息的通信信道。

按照信息传递的方式和方法区分，隐蔽通道分为存储隐蔽通道和时间隐蔽通道。存储隐蔽通道是指两个进程利用不受安全策略控制的存储单元传递信息，前一个进程通过改变存储单元的内容发送信息，后一个进程通过观察存储单元的变化来接收信息。时间隐蔽通道是指一个进程通过调整使用的系统资源（例如 CPU 时间），从而影响到实际的响应时间，另一个进程通过观察响应时间，获取相应的信息。时间信道也被称为无记忆信道，因为它无法长久地存储信息。接收者必须及时接收发送者发送的信息，否则这些信息就会消失。存储信道又称为记忆信道，因为敏感数据存储在特定存储单元中，只要该单元还存在，敏感信息就能继续存在。

为了应对隐蔽信道的威胁，在 TCSEC 标准中要求 B2 级以上的系统评估必须包括隐蔽信道分析，并且随着评估级别的升高，对隐蔽信道的分析要求也越来越严格。隐蔽信道分析的主要目的在于找到系统的漏洞，并进一步分析这些漏洞，以确定其潜在的危害。隐蔽信道分析工作包括信道识别、度量和处置。信道识别是对系统的静态分析，强调对设计和代码进行分析发现所有潜在的隐蔽信道。信道度量是对信道传输能力和威胁程度的评价。信道处置措施包括信道消除、限制和审计。隐蔽信道消除措施包括修改系统、排除产生隐蔽信道的源头、破坏信道的存在条件。限制措施要求将信道危害降低到系统能够容忍的范围内。但是，并非所有的潜在隐蔽信道都能被入侵者实际利用，如果对所有的潜在隐蔽信道进行度量和处置会产生不必要的性能消耗，降低系统效率。隐蔽信道检测则强调对潜在隐蔽信道的相关操作进行监测和记录，通过分析记录，检测出入侵者对信道的实际使用操作，为信道度量和处置提供依据。

6.11　小　　结

信息安全基础设施的关键是安全操作系统，没有操作系统的安全，就不可能真正解决数据库安全、网络安全和其他应用系统的安全问题。本章首先补充了操作系统基础知识，包括系统调用和进程的概念，在此基础上，详细介绍了操作系统的安全机制，包括操作系统的存储保护、身份认证、访问控制、审计技术等。操作系统的存储保护机制与具体采用的存储管理方式密切相关，因而本章首先详细介绍了分区式存储管理、分页存储管理、分段存储管理、虚拟存储管理等各种存储管理方式的原理，在此基础上，介绍了各种存储管理方式的存储保护手段。最后介绍了两种主流操作系统 Windows 和 Linux 所采用的安全机制。

习　　题

一、填空题

1. 威胁操作系统安全的因素主要有________、________、________。
2. 操作系统的主要功能包括________、________、________、________、________等。
3. 操作系统中最基本、最重要的概念是________。
4. 进程在执行过程中至少有三种状态：________、________、________。
5. ________是进程存在的唯一标识。
6. Linux 中采用________系统调用创建子进程。
7. 把逻辑地址转换为物理地址的过程称为________，主要有两种方式，包括________、________。
8. ________是系统审计用户操作的最基本单位。
9. 存储保护主要涉及防止________、________。
10. 虚拟分页存储管理的主要思想是________、________。
11. 在请求分页虚拟存储管理系统中，当要访问的页不在内存，系统会触发________。
12. ________是系统审计用户操作的最基本单位。
13. ________是有效实现最小特权的机制。
14. Windows 身份认证的实现主要包括________、________、________三个组件。
15. Windows 安全子系统中实现自主访问控制主要包含 5 个关键的组件：________、________、________、________、________。

二、选择题

1. Windows 主机推荐使用(　　)格式。

A. NTFS　　B. FAT32　　C. FAT　　D. Linux

2. 将 test 及其下的所有目录及文件的属主改为 test、属组改为 xmb 的命令为(　　)。

A. chown -R test:xmb test　　B. chown -R xmb:test test

C. chown test:xmb test　　D. chown xmb:test test

3. UNIX/Linux 系统中，下列命令可以将普通账号变为 root 账号的是(　　)。

A. chmod 命令　　B. /bin/passwd 命令

C. chgrp 命令　　D. /bin/su 命令

4. 保障 UNIX/Linux 系统账号安全最为关键的措施是(　　)。

A. 文件/etc/passwd 和/etc/group 必须有写保护

B. 删除/etc/passwd、/etc/group

C. 设置足够强度的账号密码

D. 使用 shadow 密码

5. Windows 系统中的审计日志不包括(　　)。

A. 系统日志(SystemLog)　　B. 安全日志(SecurityLog)

C. 应用程序日志(ApplicationLog)　　D. 用户日志(UserLog)

三、简答题

1．操作系统面临哪些安全问题？

2．操作系统安全的主要目标是什么？实现操作系统安全目标需要建立哪些安全措施？

3．Windows 系统的安全子系统组件有哪些？

4．请描述各种存储管理方式如何进行存储保护。

5．操作系统提供的安全机制有哪些？

6．什么是静态地址重定位，什么是动态地址重定位？

7．简述分页存储管理的主要思想。

8．审计的作用是什么？

9．什么是最小特权管理？

10．简述 Linux 安全机制。

11．知识扩展：访问安盟电子信息安全公司的主页 http://www.anmeng.com.cn，进一步了解身份认证产品的原理及其应用。

第7章　数据库系统安全

数据库技术从20世纪60年代产生至今，已得到快速的发展和广泛的应用。当前，信息系统大多采用数据库存储和管理大量关键数据，是攻击者攻击的主要目标。数据库系统面临的安全威胁和风险越来越大，数据库安全是信息系统安全的重要组成部分，要通过数据库管理系统DBMS的安全控制机制来实现。本章主要介绍数据库系统安全方面的知识。

本章的主要内容安排如下：7.1节、7.2节对数据库的安全性进行总体介绍，包括数据库安全的定义、数据库安全研究的发展历史和研究现状等；7.3节介绍数据库系统的身份认证技术，包括数据库中身份认证的概念和必要性、身份认证技术在数据库系统中的应用方法等内容；7.4节介绍数据库系统访问控制技术，包括关系型数据库的自主访问控制、强制访问控制和基于角色的访问控制技术；7.5节介绍数据库系统安全审计技术，包括数据库安全审计的概念和必要性、数据库安全审计的实施方法；7.6节介绍数据库系统的备份与恢复技术；7.7节介绍数据库的存储和传输加密技术；7.8节以Oracle为例介绍数据库系统中的高级安全技术；7.9节简要介绍了数据库安全评估的基本要求，发展历史和评估准则；7.10节对本章内容进行了小结。

7.1　数据库安全概述

7.1.1　数据库安全定义

目前，学术界对数据库安全尚无公认的权威定义，我国公安部的行业标准"GA/T 389-2002 计算机信息系统安全等级保护数据库管理系统技术要求"对数据库安全的定义为：数据库安全就是保证数据库信息的保密性、完整性、一致性和可用性。保密性是指保护数据库中的数据不被泄露和未授权的获取；完整性是指保护数据库中的数据不被非授权用户破坏和删除；一致性是指确保数据库中的数据满足实体完整性参照完整性和用户定义完整性要求；可用性是指确保数据库中的数据不因人为的和自然的原因对授权用户不可用。

当前，数据库面临的安全威胁主要可分为物理上的威胁和逻辑上的威胁。物理上的威胁是如水灾、火灾等造成的硬件故障，从而导致数据的损坏和丢失等。为了消除物理上的威胁通常采用备份和恢复的策略。逻辑上的威胁主要是指对信息的未被授权存取，可以分为三类：①信息泄露，包括直接和非直接(通过推理)地对保护数据的存取；②非法的数据修改，由操作人员的失误或非法用户的故意修改引起；③拒绝服务，通过独占系统资源导致其他用户不能访问数据库。为了消除逻辑上的威胁，DBMS必须提供可靠的安全策略，以确保数据库的安全性。

7.1.2　数据库安全与操作系统的关系

在数据库发展的初期，信息安全主要是依靠操作系统的文件管理功能，利用访问控制矩

阵,实现对各类文件进行的授权读写和执行等,其次,还依靠操作系统的监控程序进行用户登录和口令鉴别的控制。数据库广泛使用之后,由于数据库提供了比文件管理系统更强的功能,其共享程度更高、操作方便,而且含有重要的程序和不同级别的各类数据,因此数据安全显得尤为重要,有必要专门研究其数据保护机制。原来用于操作系统文件管理的一些保护措施可以借用到数据库的安全保护中来,但是数据库安全同操作系统安全之间存在着某些差别,例如:

(1) 数据库实行保护的对象更多,不仅仅只限于文件。

(2) 数据库中数据的生命周期通常要长一些,数据库安全涉及不同层次,如文件、数据库记录和记录中的数据项。

(3) 操作系统保护实际的资源,而数据库系统保护对象中的某些对象可能具有复杂的逻辑结构,某些对象可能反映同一物理对象。

(4) 不同的结构层,如数据库的内模式、概念模式和外模式要求有不同的安全保护。

(5) 数据库安全是涉及数据的语义以及数据的物理表示。

图 7.1 给出了数据库安全实现过程概貌图。

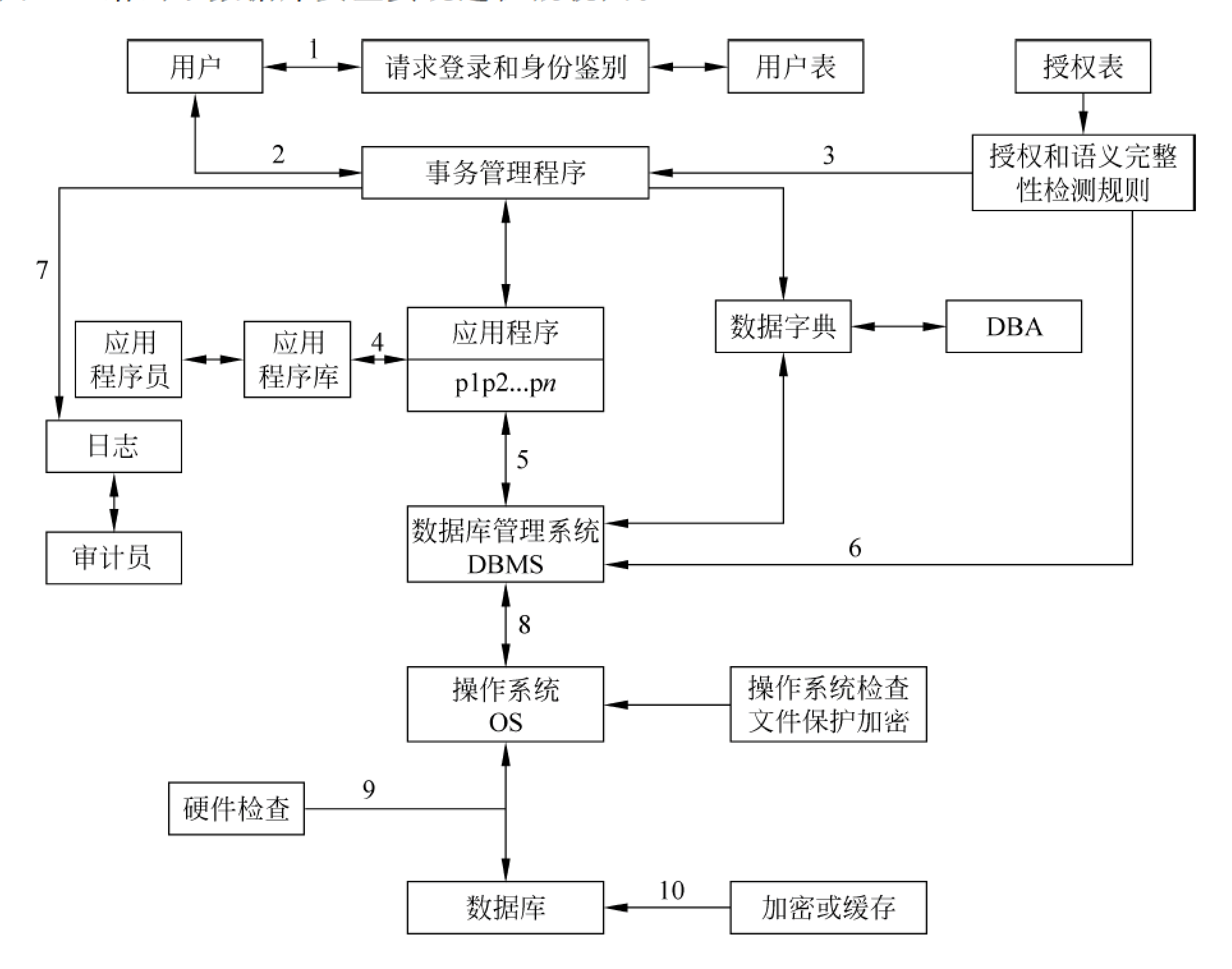


图 7.1　数据库安全实现过程概貌图

从这些差别来看,数据库的安全保护要求,要比操作系统更高更复杂。从数据库安全的角度来看,操作系统只提供文件级的管理与控制,而数据库的程度更细、控制对象更多,包括文件、记录和数据项等。因此,数据库安全所要研究的内容是十分广泛的。研究设计一个数据库系统,所涉及的安全性问题包括用户身份鉴别、事务处理存取检查、授权规则、语义完整性检查、用户登录鉴别、审计追踪、操作系统检查、文件检查、实体保护、数据库并发控制、数据恢复、统计数据库安全与推理控制以及分布式数据库安全等各个方面。本节将简要介

绍数据库安全的原理,以及实现数据库安全的重要策略和基本方法。为了对数据库安全的整体有个完整的概念,用图 7.1 来表示数据库安全实现过程的总概貌,数据库系统的事务处理安全检查过程详细说明如下。

(1) 身份鉴别及登录机制,用户请求访问时根据用户给定的标识进行识别,验证其身份。最常用的用户标识是口令,即只有用户知道的一串字母、数字的组合。用户标识也可以是 IC 卡、USB Key 等计算机系统可以识别的东西。更先进的用户标识是用户的生理特征,诸如指纹、虹膜等。鉴别机制将用户提供的标志(例如口令)与存储在用户表中的用户标识进行对照,以确定申请认证的用户是否合法。用户表中的口令文件应该用密文存放,这样可以防止有人偷窃了系统的用户表,获取明文的用户名和口令,而伪造合法用户进行登录。此外,通过身份鉴别和登录认证机制,可以确定用户以什么级别进入系统,并登记用户进入的时间。

(2) 经认证机制核查,用户可向系统申请事务处理。

(3) 由访问控制机制对用户被授权使用的事务处理进行核对其权限表,然后排队调度事务处理。

(4) 事务执行时可能要调用应用程序库中的程序。

(5) 应用程序执行时,访问请求进入数据库管理系统(Database Management System, DBMS),通过查数据字典,组织和完成对数据库的访问。

(6) DBMS 进行必要的检查,以组织对数据库的数据访问,并完成必要的授权检查。

(7) DBMS 应在执行时进一步核对事务处理请求的权限,并对并发用户的更新操作提供控制,以避免改变数据库的完整性。还要保留数据访问的每次登录信息,以供审计和恢复 DB 时使用。

(8) 请求 DB 生效后,DBMS 就按子模式到内模式、再到物理存储的映射,把访问转换成一个 I/O 请求,然后通过操作系统来完成,这期间可能要进行操作系统的检查。

(9) 对操作系统的功能,以及文件的使用做进一步核查. 并提供硬件保护,确保数据的正确传送。

(10) 最后,存放在数据缓冲里的信息可能是加密存放或是留有后备副本以作恢复之用的。

DBMS 中的安全性最终要靠操作系统和硬件设备所提供的环境实现,如果操作系统允许用户直接存取数据库文件,在 DBMS 中哪怕采取最可靠的安全措施也是没用的。为了保证数据库的安全,需要一个可靠的操作系统,操作系统应该至少提供下列安全控制功能。

(1) 保护 DBMS,防止用户程序对其进行修改,尤其是 DBMS 中的访问控制机制。

(2) 要对内存缓冲区中的数据提供保护,当敏感数据存放在内存缓冲区中时,必须防止非授权用户对其进行读写。

(3) 防止 DBMS 之外的程序对数据库直接进行存取,即除了 DBMS 之外,别的程序不能对外存中的数据库文件直接进行存取。

(4) 进行正确的物理 I/O,保证正确地读取数据库文件。

(5) 提供可靠的数据通信信道,通过通信线路传输数据时,应对其提供保护,防止泄露或被篡改。

7.2　数据库安全的发展历史

在数据库安全的理论研究中，国外的研究主要包括：

(1) 安全模型的研究及其实用化。提出了以 SeaView 模型、LDV 模型、SWORD 模型为代表的关系数据库的安全模型和原型实现。

(2) 针对数据库技术的发展，对一些非关系模型的数据库系统，如面向对象数据库系统、多媒体数据库系统的安全模型进行了研究。

(3) 对安全数据库系统中的推理控制问题进行了研究。

(4) 数据仓库的安全保密问题。

(5) 对一些特定的数据库应用系统，如美国的医疗数据库系统的特殊安全需求和实现等。

随着数据库安全理论和技术的发展，为了满足对安全数据库系统越来越迫切的需求，国外各大数据库厂商纷纷推出了各自的安全数据库产品，如 ORACLE 公司的 Trusted Oracle、Informix 公司的 Informix-online Secure、Sybase 公司的 Secure SQL Server 等产品。

基于对安全数据库的需求，国内学术界对数据库的安全问题也进行了研究，从目前掌握的资料看，国内对数据库安全的研究无论是在理论上还是在相关实用产品上都落后于国外，国外几大数据库厂商的低安全级别的数据库产品占据了国内的大部分市场。国内的电信、金融、电力、保险等行业基本上都采用的是 ORACLE、Sybase、Microsoft 三大数据库厂商和 IBM 微软的低安全级别的数据库产品。国内对数据库系统的开发基本停留在大学等研究机构的实验室内，其种类较少、应用范围也较窄，如东北大学国家软件工程研究中心开发的 OpenBase、北京大学和中国人民大学合作开发的 KingBase、华中科技大学开发的 DM 数据库系统等，其市场份额较小。

在安全数据库的理论与实践研究中，华中科技大学对数据库安全策略、安全模型、加密算法等方面进行了研究。在开发安全数据库系统方面，主要取得的成果有：

(1) 外包式的多级安全 RDBMS 实验系统，主要是利用 ORACLE RDBMS 提供的 PRO * C 和 OCI 接口实现，在扩展 ORACLE 使用的 SQL 语言和 SQL * PLUS 的基础上，构造了新的安全数据库操作语言 SSQL，含有较为完善的强制访问控制和数据加密机制。由于其仅仅是一个实验系统，存储效率不高，而且一些如索引等机制没有实现，距离实用化有相当的距离。

(2) 多级安全数据库原型 MLS DBMS 是在 UNIX 平台上进行的，目的是探索将各种数据库安全机制嵌入 DBMS 中，而 DBMS 本身的性能、功能、效率并未作过多要求。

人民大学与北京大学合作进行的国家九五项目《国产化关系数据库管理系统 Cobase V2.0》是在八五期间开发的 Cobase V1.0 的基础上进行的安全版设计，达到了美国国防部标准 B1 级，实现了安全性标识、强制访问控制和审计功能。由于 Cobase V1.0 是一个功能完备的多用户关系型数据库管理系统，具有以自主访问控制为主的安全特征。Cobase V2.0 充分利用原有系统为基础，严格遵循 B1 级标准，作为一个小型而性能完备的安全 DBMS 基本达到了实用化的程度。

DBMS能否获得广泛使用，其性能是最为重要的一点。在不降低性能或者尽可能少降低性能的基础上提高安全性，满足军队、政府等部门的高安全性要求是多级安全DBMS实用化的关键，因此若干研究机构又在另外一条途径上进行了一系列的探索，以满足军队、政府等部门的较高的安全需求，即在现有已获得广泛应用的各种商业数据库系统，如ORACLE、SYBASE、INFORMIX、MS SQL SERVER等基础上，利用原有的安全特性，将多级安全、数据加密、安全审计和强制访问控制等设计和实现技术引入管理信息系统的开发中，开发出专用的多级安全加密应用系统。在这方面进行的研究与应用主要以国防科技大学的YHOA和华中理工大学计算机系开发的，基于ORACLE 7的多级安全加密管理信息系统为代表。

(3) 原华中理工大学研制的“基于ORACLE 7的多级安全加密管理信息系统”针对军事部门的安全要求，综合考虑了系统的实用性、安全性，实现了一个“外包式”的多级安全加密管理信息系统，整个系统主要由安全控制子系统、加密子系统、文电处理子系统、审计子系统组成。

(4) 国防科技大学的YHOA系统的设计目标是办公事务的可视化管理，多媒体、网络环境下的协同工作以及异种数据库的资源共享等，以良好的图形界面为日常办公事务提供了完整的、一体化的支持，YHOA中的文档管理系统是整个系统的核心，其中的数据库子系统基本满足标示与认证、自主与强制访问控制等安全性要求。YHOA构造的数据库安全子系统与各个应用系统紧密结合，按照B1级别的安全要求设计，整个系统是在VB 3.0与ACCESS数据库上开发的，在Windows 3.2 for Workgroup环境下运行。

7.3 数据库身份认证技术

7.3.1 数据库用户身份认证概念

当用户请求进入计算机系统时，计算机操作系统首先进行标识核对，合法的用户才能进入计算机系统，但是，进入计算机系统的用户不一定具有数据库的使用权，数据库管理系统还要进一步进行身份认证，以拒绝没有数据库使用权的用户对数据库的访问。

用户身份认证是数据库系统的第一道安全防线。用户在进入数据库访问数据库资源之前，首先需要经过身份认证模块才能与数据库服务器建立连接，之后访问授权控制模块根据用户的身份和权限决定用户是否能够访问某个资源，审计系统记录用户的操作请求和行为。访问控制和审计系统都要依赖于身份认证系统的提供的“信息”——用户的身份。可见身份认证在安全系统中的地位极其重要，是最基本的安全服务，其他的安全服务都要依赖于它。一旦身份认证系统被攻破，那么系统的所有安全措施将形同虚设。

和操作系统的用户认证类似，数据库用户认证是在数据库系统前后端之间建立可信安全通信信道的重要过程，它是数据库系统的“门禁”模块，是数据库授权和审计的基础。数据库身份认证是验证用户身份与其所声称的身份是否一致的过程，其实质是用户标识的鉴别过程，是数据库系统提供的最外层安全保护措施，本书第4章已经对身份认证的过程、手段作了详细介绍，概括来说，有三个要素可以用于认证过程，即用户的知识，如口令等；用户的物品，如IC卡等；用户的特征，如指纹等。

目前，一般的数据库系统采用比较通用的用户名加口令的认证方式，符合规范的用户名和口令设置及管理，可以有效抵制非法用户的入侵。对某些针对特殊应用环境具有高度机密性要求的专用数据库系统而言，仅用上述认证方式其安全强度是不够的，比较通行可靠的方式是基于智能卡(或 USB Key)与用户标识双要素认证。认证时，用户在客户端插入智能卡(USB Key)并输入 PIN 码，客户端调用卡接口并与用户注册系统进行交互，判定是否为合法用户。这种认证方式中，卡与用户标识码两者缺一不可，否则，用户无法通过认证。与用户名加口令方式相比，这种方式提高了认证的安全性，因为用户名一般是公开信息，而智能卡是用户独有的。

除上述认证方式外，当前，基于指纹、虹膜等生物特征的认证技术也正在逐步得到推广和应用，但限于技术成本、认证精确度、认证效率等方面的问题，目前，这些技术并未广泛使用。

当前流行的 DBMS 身份认证一般采用以下几种验证方式：

1. 操作系统验证

用户进入操作系统后不需要用户名和口令而直接连接到数据库，在这种情况下，用户对数据库的连接要靠操作系统来验证，操作系统的合法用户也是数据库系统的合法用户。对 SQL Server 2000 来说，采用 Windows 认证更安全，因为 Windows NT/2000 操作系统的安全性能达到美国国防部定义的 C2 级安全标准，它的认证具有安全确认、口令加密、审核、口令有效期保护、最短口令长度限制、非法登录时的账户锁定等功能。

2. DBMS 提供验证

很多 DBMS 要求独立于操作系统进行身份认证，如 SQL Server、Oracle 等都提供了独立认证的功能。

3. 网络安全系统的认证

已经有许多网络安全认证系统可以用来对数据库用户进行认证。这要依赖于认证和密钥分配系统，用户可以通过提供身份证明或验证令牌来响应验证请求，包括采用智能卡、安全令牌、生物识别或其组合的 PKI 技术。本质上，认证和密钥分配系统提供的是一个应用编程接口(API)，它可以用来为任何网络应用程序提供安全服务，例如认证、数据机密性和完整性、访问控制以及非否认服务。比较常用的系统有 DCE、Kerberos、SESAME 等。

7.3.2　SQL Server 数据库用户身份认证机制

1. SQL Server 安全性概述

访问 SQL Server 数据库中的数据必须经过三个级别的认证过程，如图 7.2 所示。

第一个级别的认证是 Windows 级别的认证，即由操作系统进行身份验证，数据库用户必须首先是操作系统的合法用户，该级认证的主要目的是验证用户是否具有连接到 SQL Server 数据库服务器的资格。

第二个级别的认证是 SQL Server 级别的认证，该认证过程当用户访问数据库时发生，必须具有对具体数据库的访问权，即验证用户是否是数据库的合法用户。

第三个级别是数据库级，该级别是指当用户操作数据库中的数据对象时，必须具有相应的操作权，即验证用户是否具有操作权限。

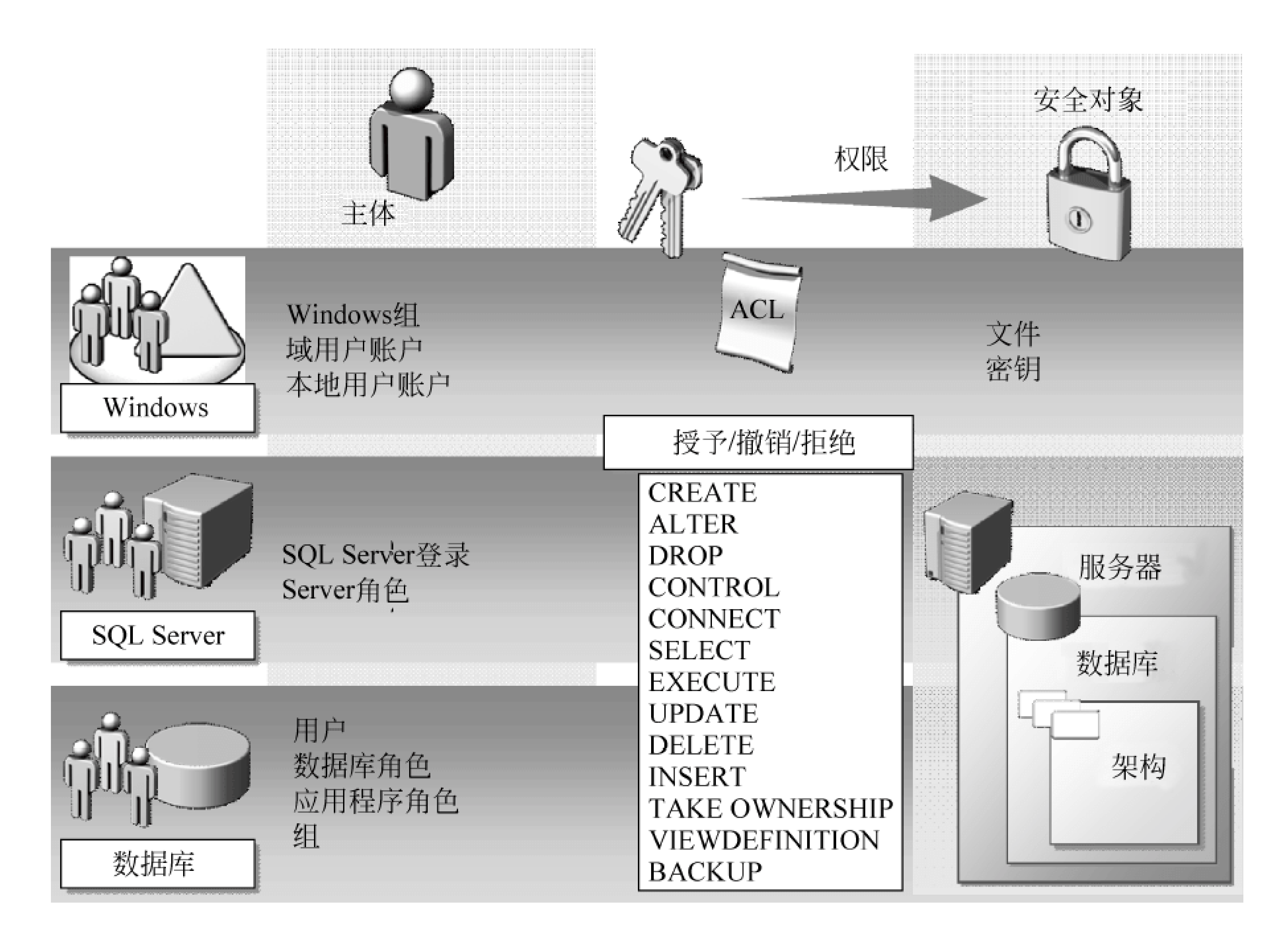


图 7.2 SQL Server 数据库三个级别的用户认证原理图

这就好比确保保存在银行保险箱中财物的安全需要多层安全措施，保险箱自身有钥匙和锁具，保险箱置于保险库中，而保险库的位置处于普通人难以到达的银行建筑的中心位置或地下，仅有通过授权的人才能进入保险库，通向保险库的道路有限且有监控系统，银行大厅有警卫巡视且有联网报警系统。通过不同层次和级别的安全措施共同保证了所存财物的安全。同样 SQL Server 数据库为了确保数据的安全设置了不同级别的认证，首先要经过操作系统的身份认证，其次要求是数据库管理系统的合法用户，最后还必须是某个特定数据库的合法用户，才能对数据库中的元素、表、视图等进行操作。

2. SQL Server 身份认证模式

SQL Server 数据库系统提供了两种身份验证模式：Windows 身份验证模式和混合模式。

使用 Windows 身份认证模式时，用户使用 Windows 操作系统中的账户名和密码连接 SQL Server，如图 7.3 所示。当用户通过 Windows 用户账户进行连接时，SQL Server 使用 Windows 操作系统中的信息验证账户名和密码，这是 SQL Server 默认的身份验证模式，比混合模式安全得多。

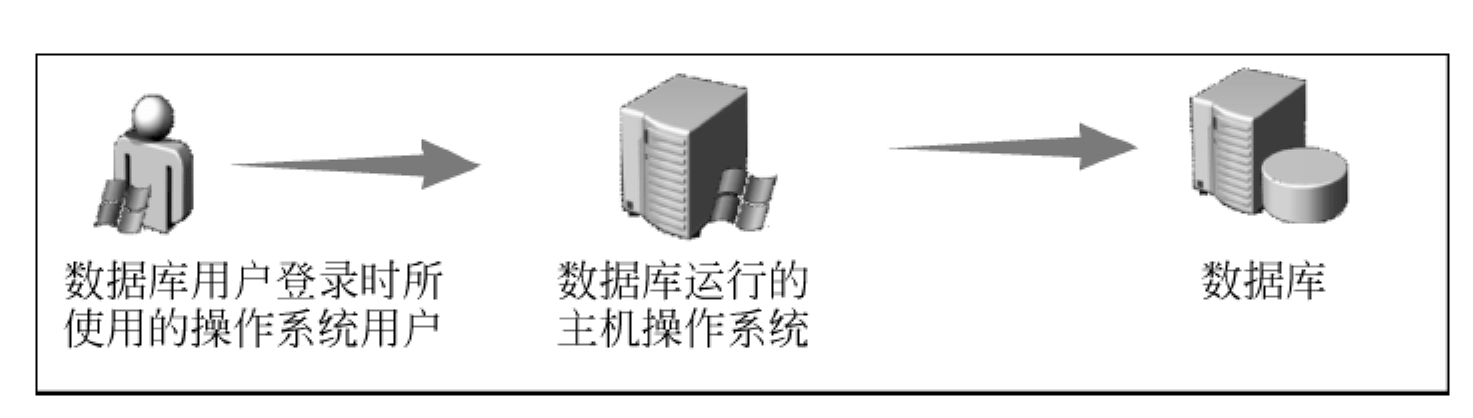


图 7.3 SQL Server 数据库 Windows 身份认证概念图

使用混合模式时，当客户端连接到服务器时，既可能采取 Windows 身份验证，也可能采取 SQL Server 身份验证，如图 7.4 所示。混合验证模式允许以 SQL Server 身份验证模式或者 Windows 身份验证模式来进行验证。使用哪个模式取决于在最初的通信时使用的网络库。如果一个用户使用 TCP/IP Sockets 进行登录验证，则使用 SQL Server 身份验证模式；如果用户使用命名管道，则登录时将使用 Windows 验证模式。这种模式能更好地适应用户的各种环境。

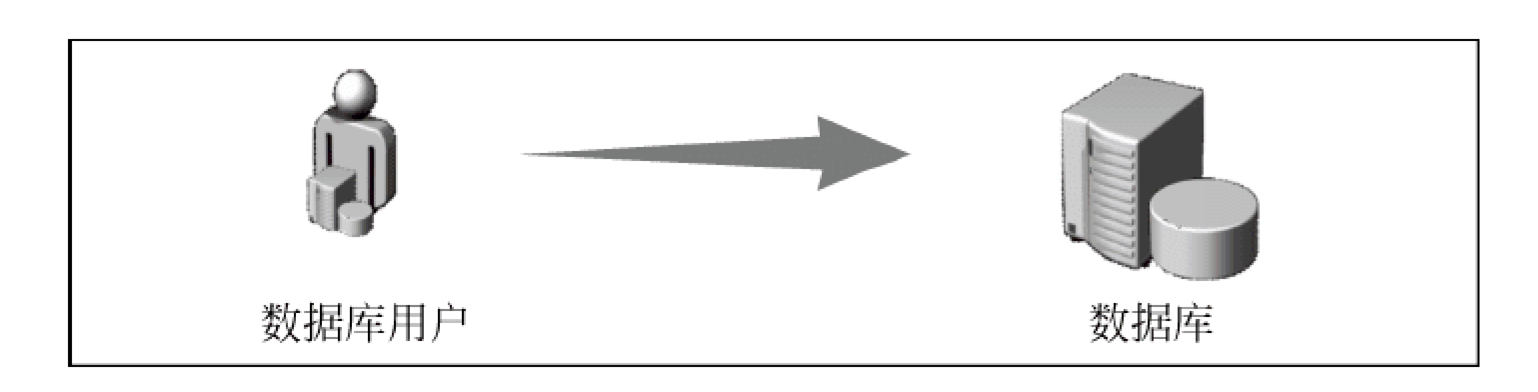


图 7.4　SQL Server 数据库混合身份认证概念图

SQL Server 的两个验证方式有明显的不同，主要集中在信任连接和非信任连接上。

1）Windows 身份验证模式

Windows 身份验证相对于混合模式更加安全，使用本连接模式时，仅仅根据用户的 Windows 权限来进行身份验证，称之为“信任连接”，但是在远程连接的时候会因 NTML 验证的缘故，无法登录。

Windows 认证模式的优点是：数据库管理员的工作集中在管理数据库方面，而不是管理用户账户，对用户账户的管理可以交给 Windows 服务器去完成。Windows 服务器有着更强的用户账户管理工具，可以设置账户锁定、密码期限等。如果不是通过定制来扩展 SQL Server，SQL Server 是不具备这些功能的。Windows 服务器的组策略支持多个用户同时被授权访问 SQL Server。该模式是默认的身份验证模式，比混合模式更为安全，在安全级别要求高的场合请尽可能使用 Windows 身份验证。

2）混合身份验证模式

混合模式验证就是当本地用户访问 SQL Server 时采用 Windows 身份验证建立信任连接，当远程用户访问时由于未通过 Windows 认证，而进行 SQL Server 认证（使用 sa 的用户也可以登录 SQL），建立“非信任连接”，从而使得远程用户也可以登录。

混合验证模式的优点是：创建了 Windows 服务器之外的一个安全层次；支持更大范围的用户，如 Novell 网用户等；一个应用程序可以使用单个的 SQL Server 登录账号和口令。

3. SQL Server 的身份验证流程

如图 7.5 所示，当用户通过 SQL Server 管理工具或数据库访问应用程序向数据库发起连接请求时，SQL Server 数据库首先读取管理员所设置的用户身份验证模式信息。如果管理员设置的是 Windows 身份验证模式，则判断当前登录的 Windows 用户是否是合法用户，如果是则接受用户发起的连接。如果管理员设置的是使用混合验证模式，则首先判断连接请求的用户是否是 SQL Server 账户，如果是则判断用户名和口令是否正确，如果正确则接受连接；如果用户不是 SQL Server 账户，则判断用户是否是合法的 Windows 账户，如果是则接受连接，否则拒绝用户的 SQL Server 数据库连接请求。

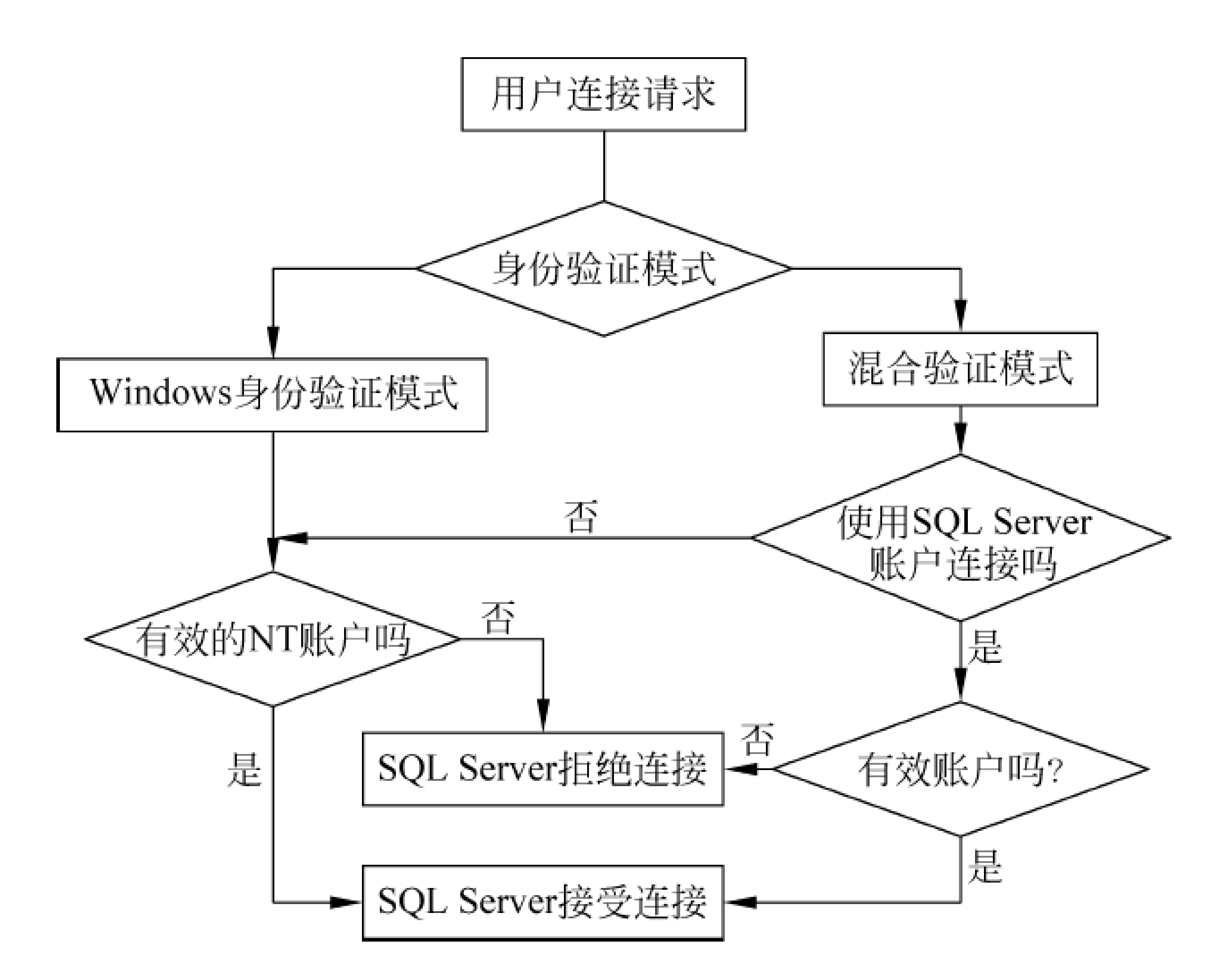


图 7.5 SQL Server 数据库身份认证流程图

7.3.3 Oracle 数据库用户身份认证机制

1. Oracle 安全性概述

Oracle 中,一个用户如果要对某一数据库进行操作,必须满足以下三个条件:

(1) 登录 Oracle 服务器时必须通过身份验证;

(2) 必须是该数据库的用户或者是某一数据库角色的成员;

(3) 必须有执行该操作的权限。

在 Oracle 系统中,为了实现这种安全性,采取了用户、角色和概要文件等管理策略。

2. Oracle 数据库用户身份认证机制

Oracle 系统有一套严格的用户管理机制,新创建的用户只有通过管理员授权才能获得系统数据库的使用权限,否则该用户只有连接数据库的权限。正是有了这一套严格的安全管理机制,才保证了数据库系统的正常运转,确保数据库信息不泄露。用户管理包含创建用户和管理用户两方面的内容。

用户是使用数据库的所有合法操作者。Oracle 数据库中有两个基本用户:SYSTEM 和 SYS,他们是系统默认创建并具有某些特权的用户。创建用户是建立一个安全、有用的账户,并且这个账户要有充分的权限和正确的默认设置值,保证用户既能通过必要的权限履行职责,但又不违反安全策略的约束。用户账户可在企业管理器(Oracle Enterprise Manager,OEM)中创建,也可使用 SQL 命令在查询分析器中创建。

对用户进行管理,是对已创建用户的信息进行管理,包括信息的添加、修改、删除。例如,为用户增加新权限、改变概要文件、创建用户报告和删除用户等。用户管理和安全紧密相关,特别是用户授权,必须符合预设的访问控制策略,通常,授权需遵循"最小特权原则",即只给用户授予履行职责任务必要的权限。

Oracle 数据库通过 sqlnet.ora 文件中的参数 sqlnet.authentication_services、参数文件中的 remote_login_passwordfile 和口令文件 pwdsid.ora 三者协同实现不同方式的身份

认证。

```
sqlnet.authentication_services = (NTS)|(NONE)
```

其中：NTS 表示操作系统认证方式，不使用口令文件；NONE 表示口令文件认证方式。

```
remote_login_passwordfile = (NONE)|(EXCLUSIVE)|(SHARED)
```

其中：NONE 表示不使用口令文件，操作系统认证；EXCLUSIVE 表示使用口令文件认证方式，但只有一个数据库实例可以使用此文件；SHARED 表示使用口令文件认证方式，允许有多个数据库实例可以使用此文件，但此设置下只有 SYS 账号能被识别，即使文件中存在其他用户的信息，也不允许他们以 SYSOPER/SYSDBA 登录。

1）操作系统认证方式

```
sqlnet.authentication_services = (NTS)并且 Remote_login_passwordfile = (NONE)
```

当以 oracle_dba 组下的用户登录进入本地的操作系统后，进行以下操作：

```
sqlplus  /nolog
SQL> conn  /as sysdba
```

可以以 sysdba 身份登录成功，进行数据库方面的操作。

当以远程进行登录时，执行：

```
sqlplus /nolog
SQL> conn /as sysdba
```

则会显示 ERROR:ORA-01031:insufficient privileges，即不允许以 sysdba 身份远程登录系统，这也是 OS 认证之所以称为本地认证方式的原因。

2）口令文件认证方式

```
Sqlnet.authentication_services = (NONE)并且 Remote_login_passwordfile = (SHARED)|(EXCLUSIVE)
```

配合口令文件 PWDsid.ora。

当在本地以 oracle_dba 组下的用户登录进入系统时，进行以下操作：

```
sqlplus /nolog
SQL> conn /as sysdba
```

则会显示 ERROR:ORA-01031:insufficient privileges。

在本地或远程进行如下操作：

```
sqlplus /nolog
```

SQL>conn sys/密码@服务名 as sysdba 可以进入系统，也就是说口令文件认证方式允许用户从本地或远程以 sysdba 身份登录，但必须提供口令。

3）操作系统认证和口令文件认证同时生效

```
Sqlnet.authentication_services = (NTS)并且 Remote_login_passwordfile = (SHARED)|(EXCLUSIVE)
```

配合口令文件 PWDsid.ora。

当在本地以 oracle_dba 组下的用户登录进入操作系统后，进行如下的操作：

```
sqlplus /nolog
SQL> conn /as sysdba
```

可以进入系统，即操作系统认证方式登录成功。

当在远程执行：

```
sqlplus /nolog
```

SQL>conn sys/密码@服务名 as sysdba 也可以正常登录到数据库系统，即口令文件认证方式登录成功。

7.4 数据库授权与访问控制技术

数据或信息安全的破坏往往是从获取访问权限开始的，因此，安全防护的重要目的之一就是防止对数据或信息的非法访问。访问控制是防止数据库非法访问的重要措施之一。

7.4.1 数据库授权和访问控制

访问控制是在用户身份得到认证后，根据授权数据库中预先定义的安全策略对主体行为进行限制的机制和手段。访问控制可分为自主访问控制(Discretionary Access Control, DAC)、强制访问控制(Mandatory Access Control, MAC)和基于角色的访问控制(Role Based Access Control, RBAC)等。

如果客体拥有者或拥有者托付对象的用户可以设置访问控制属性制约其他用户对客体的访问，这样的访问控制称为自主访问控制。自主访问控制的访问授权是基于主客体身份的。自主访问控制以分布式方式进行授权，具有很强的授权灵活性，但容易导致权限扩散，所能提供的系统安全保护等级较低。

如果对客体的访问完全由系统决定，用户个人不能更改这种控制，称这样的访问控制为强制访问控制。强制访问控制中，系统通过检查主客体的相关属性决定主体是否可以访问客体。强制访问控制以集中方式对主体进行访问授权，有利于对权限的管控，能提供较高的安全等级。

基于角色的访问控制是通过将权限指派给恰当的角色，再将角色赋予合适的用户的方式对用户进行授权控制。这样的访问控制授权方便，粒度可调节，具有很强的灵活性，有利于最小特权原则的实施，并能较好地契合实际环境的需求。

数据库访问控制的前提是对用户授权，数据库系统中，不同用户由于其职责可信度的不同，能获得权限的情况也不相同。在用户分等级设置的系统中，高等级用户所能获得的权限要多于低等级用户；在用户分职责设置的系统中，用户依“最小特权原则”进行授权，即用户只能获得其职责范围内必需的权限。

1. 用户的访问权限

用户对某一数据对象的操作权力称为权限。在 DBMS 中，用户的访问权限由两个要素

组成：数据库对象和操作类型。

(1) 数据库对象包括数据库、基本表、表中记录、属性值、视图、索引等。

(2) 操作类型包括 SELECT、INSERT、DELETE、UPDATE、REFERENCES、ALL PRIVILEGES。其中，ALL PRIVILEGES 是所有权限的简写形式，REFERENCES 表示允许用户定义新关系时，引用其他关系的主键作为外键。

一般 DBMS 将数据库用户分为如下 4 类。

(1) 系统管理员用户。在一个 DBMS 上拥有一切权限的用户，如同操作系统中的超级用户，负责整个系统的管理，可以建立多个数据库，在所有数据库上拥有所有权限。一般 DBMS 在安装时至少有一个系统管理员用户。例如，SQL Server 默认的系统管理员用户是 sa，负责该 SQL Server 上的所有系统管理。

(2) 数据库管理员用户(DBA)。在某一数据库上拥有一切权限的用户，负责一个具体数据库的建立和管理。在 SQL Server 中称作 dbo(Database Owner)，也称作数据库属主或数据库拥有者。

(3) 数据库对象用户。可以建立数据库对象(如表、视图等)的用户，在自己建立的数据库对象上拥有全部操作权限。在 SQL Server 中称为 dboo(Database Object Owner)，也称作数据库对象属主或数据库对象拥有者。

(4) 数据库访问用户。一般的数据库访问用户，可以对被授权的数据库对象进行操作(如查询数据、修改数据等)。

这 4 类用户的权限逐渐降低，一般较低级别用户的权限是较高级别用户授予的。如系统管理员用户授权某个用户可以建立数据库，则该用户便可以建立数据库，并成为建立数据库的数据库管理员用户。

2. 权限的授予与收回

目前大部分 DBMS 支持自主访问控制，我们主要讨论基于自主访问控制的授权控制。DBMS 通过 SQL 提供的 GRANT 和 REVOKE 语句定义用户权限，形成授权规则，并将其记录在数据字典中。当用户发出访问数据库的操作请求后，DBMS 授权子系统查找数据字典，根据授权规则进行合法性检查，以决定接受还是拒绝执行此操作。

1) GRANT 语句

GRANT 语句用于向用户授予权限，其一般格式为：

```
GRANT <权限列表> ON <数据库对象>
  TO <用户列表>
    [WITH GRANT OPTION];
```

选项 WITH GRANT OPTION 表示被授权的用户可以将这些权限继续转授给其他用户。

【例 7-1】 现有表学生(学号，姓名，所在系，性别)，把查询学生表和修改学生学号的权限授给用户 U4，并允许其将权限转授出去。

```
GRANT   SELECT,UPDATE(学号)   ON 学生
  TO   U4
    WITH GRANT OPTION;
```

2）REVOKE 语句

权限可以由 DBA 或其他授权者用 REVOKE 语句收回，其一般格式为：

```
REVOKE[GRANT  OPTION  FOR] <权限列表>
  ON<数据库对象>
    FROM <用户列表>
      [RESTRICT|CASCADE];
```

选项 GRANT OPTION FOR 表示是部分授权权限被收回；CASCADE 表示把该用户所转授出去的权限同时收回；RESTRICT 表示限制级联收回，即只有当用户没有给其他用户授权时，才能收回权限，否则，系统拒绝执行授权动作。

【例 7-2】 收回用户 U4 修改学生学号的权限，并级联收回所授出的权限。

```
REVOKE  UPDATE(学号)
  ON  学生
    FROM  U4
      CASCADE;
```

7.4.2 SQL Server 数据库权限和角色机制

数据库系统中不能缺少用户身份认证机制，但仅提供这样的机制是远远不够的，还必须有相配套的用户权限管理机制。只有给用户分配了合适的权限，才能保证登录系统的合法用户在各自职责范围内进行操作。

SQL Server 数据库中，权限可以分为两个方面：一个是对数据库服务器自身的控制权限，如创建、修改、删除数据库，管理磁盘文件，添加、删除连接服务器等；另一个是对数据库数据方面的控制权限，如可以访问数据库中的哪些数据表、视图、存储过程等，或是对数据表执行哪些操作，是 INSERT，还是 UPDATE，或是 SELECT 等。

SQL Server 数据库管理系统可以利用角色设置，管理用户的权限。这样只对角色进行权限设置便可以实现对该角色中所有用户权限的设置，大大减轻了管理员的工作量。

SQL Server 数据库具有固定服务器角色、固定数据库角色、用户自定义数据库角色三种不同类型的角色，如图 7.6 所示。

角色
- 固定服务器角色
- 数据库角色
  - 固定的数据库角色
  - 用户定义的数据库角色

图 7.6 SQL Server 数据库角色图

服务器角色独立于各个具体的数据库，主要用来进行服务器的管理。根据 SQL Server 管理任务，把具有管理职能的用户划分为不同的用户组，每一组定义为一种固定服务器角色，如图 7.7 所示。



图 7.7 SQL Server 数据库固定服务器角色原理图

SQL Server 数据库中常用的固定服务器角色如表 7.1 所示。

表 7.1　SQL Server 固定服务器角色列表

角　色	描　述
sysadmin	可执行任何操作
dbcreator	创建和修改数据库
diskadmin	管理磁盘文件
serveradmin	配置服务器级的设置
securityadmin	管理和审核服务器登录
processadmin	管理 SQL Server 进程
bulkadmin	执行 BULK INSERT 语句
setupadmin	配置和复制已链接的服务器

固定数据库角色是将常用的数据库操作权限集合划分为不同的组，每一组称为一个固定数据库角色。SQL Server 数据库中常用的固定数据库角色如表 7.2 所示。

表 7.2　SQL Server 固定数据库角色列表

角　色	描　述
db_owner	数据库所有者
db_accessadmin	数据库访问权限管理者
db_securityadmin	数据库安全管理员
db_ddladmin	数据库 DDL 管理员
db_backupoperator	数据库备份操作员
db_datareader	数据库数据读取者
db_datawriter	数据库数据写入者
public	每个数据库用户都是 public 角色的成员

用户可以在数据库级别上，也可以对特定数据库对象定义角色，称为用户自定义角色。

7.4.3　Oracle 数据库权限和角色机制

Oracle 数据库角色分两种，一种是服务器角色，另一种是数据库角色。

服务器角色是系统预定义的，用户不能创建新的服务器角色，也不能改变服务器角色的权限，只能选择合适的已固定的服务器角色。角色管理包括修改角色的权限、生成角色报告和删除角色等工作。

由于使用角色管理权限比较简单，所以一般先将权限授予角色，然后分配给各个用户。Oracle 支持系统权限和方案对象权限。

1. 系统权限

系统权限是执行特定操作(例如创建数据库、从表中删除行数据等)的权限。Oracle 中有 60 种不同的系统权限。系统权限可以被授予用户和角色，如果将系统权限授予某个角色，就可以使用该角色管理系统权限。有两种方式可以授予或回收系统权限，一是使用 OEM，二是使用 SQL 语句 GRANT 和 REVOKE。

2. 方案对象权限

对象权限是对特定对象执行特定操作的权利，这些对象主要包括表、视图、序列、过程、

函数和包等。有些对象(如簇、索引、触发器和数据库链接)没有对应的对象权限,它们是通过系统权限控制的。例如,修改簇用户必须拥有 ALTER ANY CLUSER 系统权限。Oracle 对象有以下 9 种权限。

(1) SELECT:读取表、视图、序列中的行。

(2) UPDATE:更新表、视图和序列中的行。

(3) DELETE:删除表、视图中的数据。

(4) INSERT:向表和视图中插入数据。

(5) EXECUTE:执行类型、函数、包和过程。

(6) READ:读取数据字典中的数据。

(7) INDEX:生成索引。

(8) REFERENCES:生成外键。

(9) ALTER:修改表、序列、同义词中的结构。

7.5 数据库安全审计技术

随着数据库应用规模越来越大,系统中的用户越来越多,需要提供各类安全机制以有效控制每个用户权限、防止用户非法操作数据。但是,任何数据库系统的安全保护措施都不是完美无缺的,蓄意盗窃、破坏数据库的人总是想方设法打破控制。所以,只有安全防护是不够的,还需要能够时刻了解数据库的使用状况,并在出现问题时能及时地发现问题所在,以便解决问题、避免和挽回不必要的损失,这就需要数据库系统有一个完备的审计系统。

7.5.1 数据库安全审计定义、地位和作用

审计作为一种安全保障机制,在最早的 TCSEC 中就已有了明确的要求,从 C2 级开始,TCSEC 就制定了详细的审计功能要求,并要求信息系统对审计数据进行查询、监视,以发现对系统发起的攻击和系统的安全漏洞。审计是数据库安全的有机组成部分。

审计就是收集、记录与系统安全有关的活动,并对其进行分析处理、评估审查,查找系统的安全隐患,对系统安全进行审核、稽查和计算,追查造成安全事故的原因,并做出进一步的处理。数据库审计功能将用户对数据库的每一次更新操作(包括事务开始、事务结束以及对数据库的插入、删除、修改等)自动记录下来,便于调查,追踪责任人。审计记录一般包括以下内容:终端标识符,用户识别符,处理类型,数据更新前、后的值等。审计记录被存放在日志文件中,DBA(Database Administrator,数据库管理员)可以利用日志中的审计记录信息,重现导致数据库状况的一系列事件,找出非法访问数据的人、时间和内容等。审计是很费时间和空间的,所以审计功能一般用于安全性较高的系统中。

7.5.2 数据库安全审计方法

1. 数据库审计系统模型

一个数据库审计系统的模型包括两个部分:审计数据采集器,用于采集审计数据;审计数据分析器,负责对审计数据采集器发送给它的数据进行分析。如图 7.8 所示为安全审

计系统模型的框图，其中审计数据字典描述了系统需要审计的事件，是审计策略的体现，采集器根据审计数据字典采集数据，并存储为日志。

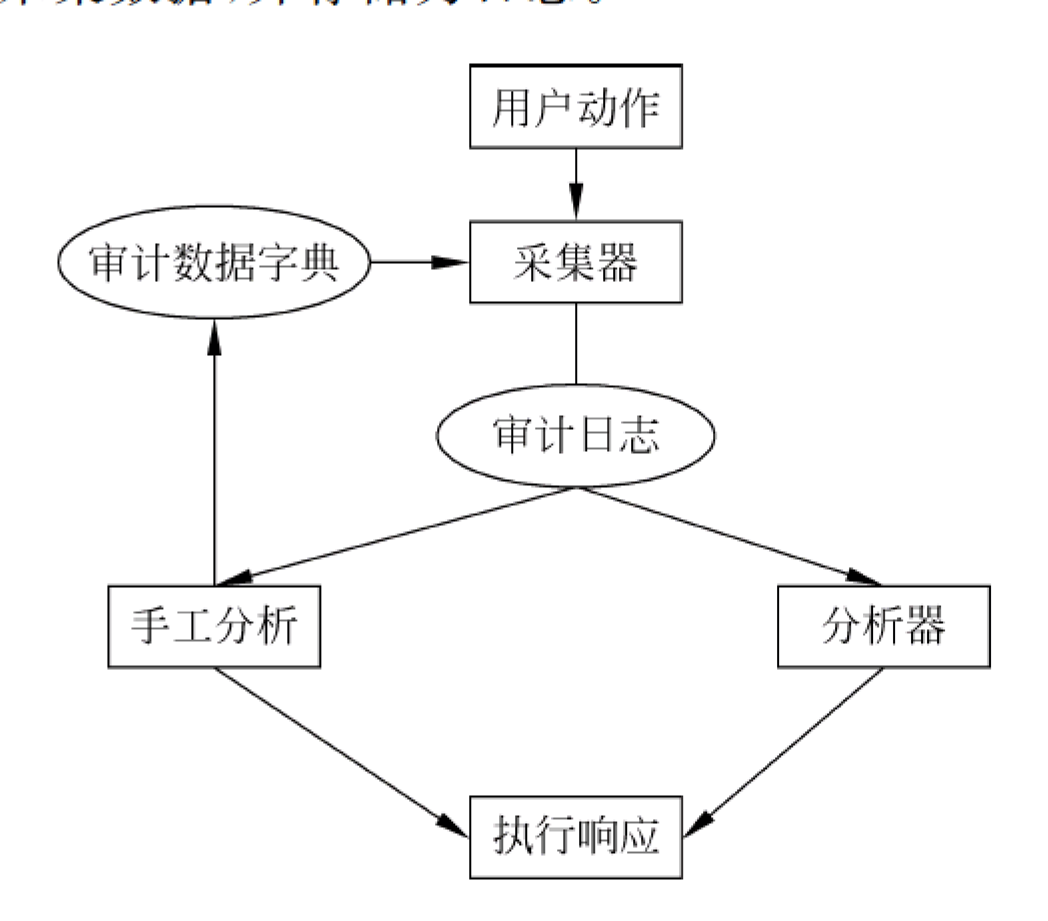


图 7.8　数据库安全审计系统模型框图

数据库安全审计系统首先收集来自用户的事件，例如数据库查询、插入、删除、修改操作等，根据相应的审计条件(审计数据字典)，判断是否是审计事件。对审计事件的内容按日志的模式记录到审计日志中。当审计事件满足报警条件时，分析器则向管理人员发送报警信息并记录其内容。当事件在一定时间内连续发生、满足逐出系统条件时，则将引起该事件的用户逐出系统并记录其内容。数据库管理员也可以通过手工分析的形式查询、检查审计日志以形成审计报告。检查的内容包括审计事件类型、事件安全级、引用事件的用户、报警、指定时间内的事件以及恶意用户表等，上述内容可结合使用。当发现新的具有潜在危害性，而审计数据字典未记录的操作，管理员可以向数据字典中更新该类型操作。

2. 审计范围与日志格式

在模型中，审计日志起着非常关键的作用，它记录了各种类型的事件，为数据库管理人员提供了事后审计的依据，掌握用户(攻击者)的企图，日志也可以作为日后用作反抵赖的具有说服力的证据。

审计日志一般应该包括发生的时间(包括日期及时、分、秒)、主机名、进程 ID 号、用户名、事件名称、出错代号、附加信息等。不同的应用系统审计日志的格式不同。广泛认同的标准格式将有利于克服非兼容性和互操作性，而非兼容性和互操作性是审计数据分析系统的开发者所面临的主要问题。采用标准的格式也有利于来自不同审计系统的审计数据的交换，并促进网络环境下对数据的协同分析。关于审计追踪日志的格式，有以下两种标准：

(1) Bishop 的标准审计日志格式。在此标准格式中，每个日志记录包含一些域，域之间由域分隔符“＃”分开，由启动和终止符号“S”和“E”来定界。域的数目是不固定的，以满足扩展性的需要。全部的数值都是 ASICII 代码串，这就避免了字节排序和浮点格式的问题。下面是一个使用 Bishop 标准审计日志格式存储的数据库审计日志的例子。

【例 7-3】 某数据库安全审计系统使用 Bishop 格式存储审计日志，它的一条日志为：

```
S20140706 18:12:13＃210.15.1.145＃teachsystem＃system＃SELECT＃teachsystem.score＃
1＃permission deniedE
```

上述日志的含义是：2014 年 7 月 6 日 18 时 12 分 13 秒，system 用户在 210.15.1.145 主机上通过 teachsystem 进程对 teachsystem.score 表进行了 SELECT 操作，结果是失败的 (0 表示成果，1 表示失败)，失败原因是“授权不允许”(Permission Denied)。

(2) 归一化的审计数据格式。归一化的审计数据格式 NADF(Normalized Audit Data Format)是由 ASAX(Advanced Security Audit-trail Analysis on UNIX)误用检测系统的开发者所定义的，旨在提供一定程度的操作系统独立性。NADF 审计追踪是有序的 NADF 记录文件，任何审计追踪都能转化成 NADF 格式。在转换时，对本地审计追踪的审计记录被抽象成为一系列审计数据值。每个审计数据值存放在一个独立的 NADF 记录中。一个 NADF 文件是有 NADF 格式的记录序列组成的文件。每条记录包括下列域：

一个 4 字节的证书，表示整个 NADF 记录的长度(包括长度域本身)；一系列连续的审计数据域，每个审计数据域包括如下三个相邻的项。

① 标识符：一个 unsigned short(16 位)类型的整数用于标识审计数据；

② 长度：一个 unsigned short 整数，标识审计数据值的长度；

③ 值：用于记录审计数据的内容本身。

图 7.9 显示了一个一般的 NADF 记录的布局。

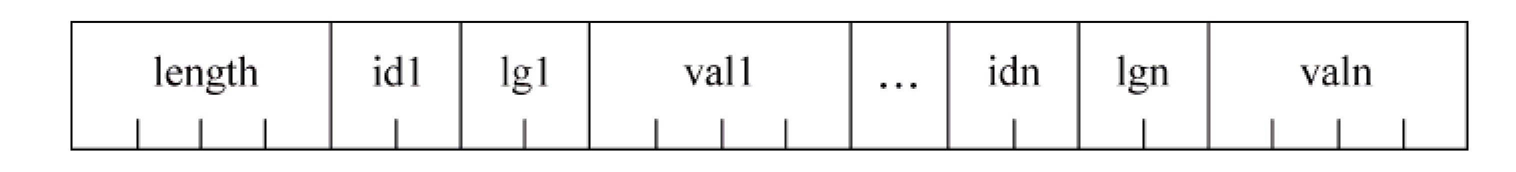


图 7.9 NADF 记录格式图

3. 安全审计日志的分析方法

审计日志需要进行分析以确定脆弱性，建立可计算性，评估损失和恢复系统运行。审计日志分析方法有人工分析和采用分析工具进行分析两种。由于日志中一般包含的数据量比较大，为了提高分析效率，可借助于审计分析工具，在开发有效的审计分析工具时，所遇到的主要障碍是需要处理日志机制生成的大量数据，这些工具将审计数据作为输入，而把审计分析后所生成的结果作为输出。

审计分析工具主要基于两种分析方法：统计分析和基于规则的专家系统。

统计分析方法定期收集与合法用户行为有关的数据，而后对观察的行为进行统计检验，以高可信度决定是否与合法用户行为相符。统计分析又称为异常检测或基于行为的检测，这种模型的特点是首先总结正常操作应该具有的特征，例如特定用户的数据库操作习惯与某些数据库操作的频率等；在得出正常操作的模型之后，同时检测从审计日志报告上来的当前活动，将它们与正常的活动比较。异常检测试图用定量方式描述可接受的行为特征，来区分非正常的、潜在的入侵行为。一旦发现正在进行监视的目标数据库事务偏离统计学意义上的正常操作模式，即进行报警。由于这种方法的数据来源是正常操作的审计记录，而历史记录经常不能包括所有的用户正常模型，因此误报警率比较高。

基于规则的专家系统采用了不同的方法，这些系统与“异常检测”的区别是：通过采用预先设定的规则来进行“滥用检测”，而这些规则都是由数据库管理员预先设计好的。这种模型的特点是收集非正常操作也就是入侵行为的特征。检测时判别所收集到的数据特征是否在入侵模式库中存在。此方法类似于杀毒软件，对事先定义的入侵特征能够准确检测，所以较少有误报警。但是，由于这种检测是基于非法操作模型，如果出现了非法操作集中没有

包括的攻击,误用检测就无能为力了,就会造成漏报警。

7.5.3　Oracle 数据库安全审计技术

Oracle 审计机制为我们提供了监视和记录数据库活动的功能,通常用于监视重要的数据库活动和收集特定的数据库活动信息。利用该特性,我们可以实现对数据库系统的操作行为进行记录以及对特定重要业务数据表的控制,甚至可以实现应用系统的操作与数据库动作的关联。

1. Oracle 数据库安全审计定义

Oracle 数据库提供的审计功能分为用户级审计和系统级审计。用户级审计是任何 Oracle 用户可设置的审计,主要是用户针对自己创建的数据库表或视图进行审计,记录所有用户对这些表或视图的一切成功和(或)不成功的访问要求以及各类型的 SQL 操作。系统级审计只能由 DBA 设置,用以监测成功或失败的登录要求、监测 GRANT 和 REVOKE 操作以及其他数据库级权限下的操作。Oracle 的审计功能很灵活,是否使用审计、对哪些表进行审计、对哪些操作进行审计等都可以根据用户的需要进行选择。

Oracle 数据库自身的安全审计机制,能够对发生在数据库里的所有操作进行审计,产生的审计记录既可以写到操作系统的审计跟踪文件里,也可以写到 SYS. AUD$ 表里,审计记录包括被审计的操作、执行操作的用户、操作的时间、操作的类型等。

2. Oracle 数据库安全审计功能开启

审计数据的产生必须要数据库开启了审计功能。Oracle 数据库的安全审计功能在默认情况下是处于关闭状态的,因为审计功能的开启将会从空间和性能两个方面对系统产生影响。因此,要想获得 Oracle 数据库的审计记录,我们要做的第一步工作是开启数据库审计功能。在 Oracle 数据库中,要想激活某个数据库上的审计功能,这个数据库的初始化参数文件里面就必须包含 audit_trail 参数。audit_trail 的作用是启用或禁止数据库审计,取值范围为 NONE、FALSE、DB、TRUE 或 OS。其中,NONE 和 FALSE 表示关闭 Oracle 审计功能;DB 和 TRUE 表示开启审计功能,并且审计日志记录在 SYS. AUD$ 表中;OS 表示开启审计功能,并且审计日志记录在操作系统的文件系统中。

3. Oracle 数据库安全审计策略配置和审计日志查看

Oracle 数据库的审计机制可以实现对三种不同的操作类型进行审计:登录企图、对象访问和数据库操作。下面将分别介绍三种操作类型的具体内容。

(1) 登录审计:用户连接数据库的操作过程称为登录,攻击者常常会采用猜测口令的方式来尝试登录到各种账户上去,Oracle 的登录审计将对每个连接数据库的企图进行审计,记录下成功、不成功,或者全部的登录企图。

与登录审计相关的命令包括:

① AUDIT SESSION;

② AUDIT SESSION WHENEVER SUCCESSFUL;

③ AUDIT SESSION WHENEVER NOT SUCCESSFUL;

④ NOAUDIT SESSION。

第 1 条命令开启连接数据库审计,第 2 条是只审计成功的连接,第 3 条是只审计失败的

连接，第 4 条命令禁止登录审计。

数据库的审计记录存放在 SYS 方案中的 AUD$ 表中，但是直接查看该表获取数据库审计记录太过复杂，Oracle 为了方便管理员查看各种类型的审计日志记录，定义了很多“简化”的视图，可以通过这些视图方便地查看 Oracle 的审计日志。登录审计日志可以通过 DBA_AUDIT_SESSION 视图来查看。

【例 7-4】 查看 Oracle 用户登录审计日志的 SQL 语句。

```
SELECT OS_Username,Username,Terminal,DECODE(Returncode,'0','Connected',
'1005','FailedNull','1017','Failed',Returncode),TO_CHAR(Timestamp,'DD-MON-YY HH24:MI:SS'),TO_
CHAR(Logoff_time,'DD-MON-YY HH24:MI:SS')
FROM DBA_AUDIT_SESSION;
```

(2) 操作行为审计：Oracle 可以对影响到某个数据库对象(例如表、表空间、同义词、数据库链接、回退段、用户或索引)的一切操作进行审计。操作行为审计的语法格式为：

```
AUDIT {statement_opt|system_priv}
[BY user, … n]
[BY {SESSION|ACCESS}] [WHENEVER [NOT] SUCCESSFUL]
```

① statement_opt：审计操作。对于每个审计操作，产生的审计记录含有下述信息：执行操作的用户、操作类型、操作涉及到的对象及操作的日期和时间。审计记录被写入审计跟踪(Audit Trail)，审计跟踪包含审计记录的数据库表。可以通过数据字典视图检查审计跟踪来了解数据库的活动。

② system_priv：指定审计的系统权限。Oracle 可对指定的系统权限和语句选项组进行审计。

③ BY user，…*n*：指定审计的用户。若忽略该子句，Oracle 审计所有用户的语句。*n* 表示可同时指定多个用户。

④ BY SESSION：同一会话中同一类型的全部 SQL 语句仅写单条记录。

⑤ BY ACCESS：每个被审计的语句写一条记录。

(3) 对象审计：Oracle 不仅能对数据库对象上的系统级操作行为进行审计，还可以对数据库对象上的数据操作行为进行审计。对象审计用于对特定数据库对象的操作进行审计记录。对象审计的语法格式如下：

```
AUDIT {object_opt|ALL} ON
{[schema.]object|DIRECTORY directory_name|DEFAULT}
[BY SESSION|ACCESS]
[WHENEVER [NOT] SUCCESSFUL]
```

① object_opt：指定审计操作。

② ALL：指定所有对象类型的对象选项。

③ schema：包含审计对象的方案。若忽略 schema，则对象在自己的模式中。

④ object：标识审计对象。对象必须是表、视图、序列、存储过程、函数、包、快照或库，也可以是它们的同义词。

⑤ ON DEFAULT：缺省审计选项，以后创建的任何对象都自动用这些选项审计。用于视图的缺省审计选项总是视图基表的审计选项的联合。

⑥ ON DIRECTORY directory_name：审计的目录名。

⑦ WHENEVER SUCCESSFUL：只审计完全成功的 SQL 语句。

(4) 权限审计。权限审计表示只审计某一个系统权限的使用状况。既可以审计某个用户所使用的系统权限，也可以审计所有用户使用的系统权限。

【例 7-5】 分别对 nick 和 admin 用户进行系统权限级别的审计。

```
SQL> audit delete any table whenever not successful;
SQL> audit create table whenever not successful;
SQL> audit alter any table,alter any procedure by nick by access Whenever not successful;
SQL> audit create user by admin whenever not successful;
```

通过查询数据字典 DBA_PRIV_AUDIT_OPTS(必须以 sys 用户连接数据库进行查询)，可以了解对哪些用户进行了权限审计及审计的选项。

```
SQL> SELECT USER_NAME,PRIVILEGE,SUCCESS,FAILURE 2
    FROM DBA_PRIV_AUDIT_OPTS
    ORDER BY USER_NAME;
```

7.6　数据库备份与恢复技术

尽管数据库系统采取各种保护措施来防止数据库的安全性和完整性被破坏，但是计算机系统受到水灾、火灾等自然灾害，硬件的故障，软件的错误，操作员的失误以及恶意的破坏仍是不可避免的，这些故障轻则造成运行事务非正常中断，影响数据库数据的正确性，重则破坏数据库，使数据库中的全部或部分数据丢失，从而使数据库处于错误状态。数据库系统需要建立一整套备份与恢复机制，及时恢复系统中的重要数据，尽可能地避免数据损失，使数据库正常运行。

备份和恢复是两个互相联系的概念，备份是将数据信息保存起来；而恢复则是当意外事件发生或者有某种需求时，将已备份的数据信息还原到数据库系统中。

7.6.1　数据库备份技术

在数据库系统中，为了保证在多用户共享数据库以及系统发生故障后仍能保证数据库中数据的正确性，引入了事务的概念，事务是一组需要一起执行的操作序列，是数据库系统的逻辑工作单元，事务具有原子性、持久性、一致性、隔离性四大特性。

(1) 原子性(Atomicity)：事务中包括的诸操作要么都做，要么都不做。

(2) 一致性(Consistency)：事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。

(3) 隔离性(Isolation)：一个事务的执行不能被其他事务干扰。

(4) 持久性(Durability)：一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。

备份是应对故障的有效手段，首先讨论一下数据库故障的类型。

1. 故障的类型

数据库可能发生的故障有以下几类。

1）事务内部的故障

如事务执行过程中发生运算溢出（试图用0作除数）、并发事务发生死锁而被选中撤销、违反了某些完整性限制等。此时，系统会强迫发生故障的事务终止运行。夭折事务的部分执行结果可能已经更新到数据库中，破坏事务的原子性。

2）系统故障

系统故障是指造成系统停止运转的任何事件，如特定类型的硬件错误（CPU故障）、操作系统故障、DBMS代码错误、突然停电等，发生系统故障后，系统要重新启动，由于内存是“易失性的”，会造成主存内容，尤其是数据库缓冲区中的内容丢失，所有故障发生时正在运行的事务非正常终止，但不会影响磁盘上的数据库。

发生系统故障后，一些夭折的事务的部分执行结果可能已写入磁盘上的数据库；有些已经提交的事务对数据库的更新结果可能还在缓冲区中，未来得及写回磁盘物理数据库中，因此系统故障破坏了数据库的原子性和持久性。

3）介质故障

介质故障又称为硬故障（Hard Crash），通常指外存故障，如磁盘损坏、磁头碰撞、瞬时强磁场干扰以及由于地震、爆炸等灾难性事件发生而引发存储介质完全毁坏等，这类故障将破坏数据库，并影响正在存取这部分数据的所有事务。

发生介质故障后，会破坏磁盘上的物理数据库，导致已提交事务对数据库的更新结果丢失，并影响正在存取这部分数据的所有事务。因此介质故障会破坏事务的原子性和持久性。

为了实现故障恢复，采用的技术是备份，不仅要备份数据库，而且还要对事务的更新操作进行备份以助恢复事务故障和系统故障。

2. 备份技术

1）日志

DBMS维护了一个日志（Log）文件来记录事务对数据库的更新操作，以助事务的恢复。日志文件的内容包括事务的开始标记（BEGIN TANSACTION）、事务的结束标记（COMMIT 或 ROLLBACK）、事务的所有更新操作。对于更新操作的日志记录，包括如下信息：事务的标识（标明是哪个事务）、操作的对象（记录内部的标识）、更新前数据的旧值（对插入操作而言，此项为空值）、更新后数据的新值（对删除操作而言，此项为空值）。具体日志记录形式如下。

[start_transaction，T]：事务T开始执行。

[write，T，A，旧值，新值]：事务T已将数据项A的值从旧值改为新值。

[commit，T]：事务T成功完成，其结果已被提交（永久记录）给数据库。

[Abort，T]：事务T异常中止，已撤销对数据库的更新。

引入日志后，每一个数据库更新操作实际上涉及两步操作：执行更新数据库的操作，将更新操作记录到日志中，这两步操作执行的先后顺序对系统有影响吗？如果先更新数据库

后写日志，由于有可能在这两步操作之间发生故障，这样的执行顺序就无法恢复更新操作了，因此需要遵循“日志先写”的原则，即必须先将更新操作记录到日志中，而后执行更新操作。

2）数据备份

日志可以提供针对事务故障和系统故障的数据恢复，为了在发生介质故障造成磁盘上数据丢失时也能进行数据库恢复，通常还需要采用数据备份技术。

DBA 定期地将整个数据库复制到另一个磁盘或磁带上保存起来，根据备份时系统状态的不同，转储可分为静态备份（冷备份）和动态备份（热备份）。

（1）静态备份是在系统中无运行事务时进行的备份，这种备份方法简单，并且能够得到一个一致性的副本，但会降低数据库的可用性。

（2）动态备份是指备份期间允许对数据库进行存取或修改，即备份和用户事务可以并发执行。动态备份可以克服静态备份的缺点，它不用等待正在运行的用户事务结束，也不会影响新事务的运行。但是，备份结束时后援副本上的数据并不能保证正确有效。

7.6.2　数据库恢复技术

下面讨论如何利用日志和数据库备份来实施数据库恢复，将数据库恢复到故障前的某个一致性状态。针对不同故障，恢复的策略也不同。

1. 事务故障的恢复

事务故障导致事务非正常终止，夭折事务的部分执行结果可能已更新到物理数据库中，破坏了事务的原子性。

恢复子系统要在不影响其他事务运行的情况下，强行回滚（ROLLBACK）该事务，具体做法是：利用日志文件撤销（UNDO）此事务已对数据库进行的修改，使得该事务好像根本没有启动一样。通常做法是逆向扫描日志文件，对于日志中记录的事务的更新操作，将更新前的值写入数据库。

2. 系统故障的恢复

发生系统故障后内存中数据库缓冲区的内容都将丢失，所有运行事务都非正常终止，这将导致一些尚未完成事务的部分更新执行结果已经写入磁盘上的数据库，而有些已完成事务对数据库的更新还留在缓冲区中，尚未写回磁盘上的数据库中，从而造成数据库可能处于不正确的状态。

恢复子系统必须在系统重新启动时，让所有非正常终止的事务回滚，强行撤销（UNDO）所有未完成事务，以保持事务的原子性，重做（REDO）所有已提交的事务以保持事务的持久性，从而保证数据库恢复到一致性的状态。重做的具体过程为正向扫描日志，对于事务的每一个更新记录，将更新后的值写入数据库。

为了提高系统故障恢复的效率，可采用具有检查点（Checkpoint）的系统故障恢复技术。

3. 介质故障的恢复

介质故障破坏物理数据库，使得所有已提交的事务的结果不能持久保存，并影响正在存取这部分数据的事务，破坏事务的原子性和持久性，是最严重的一种故障。

介质故障的恢复不仅要使用日志，还要借助于数据库备份。对于静态备份，装入数据库

备份后即处于一致性状态，利用日志重做故障前已经完成的事务，就能将数据库恢复到故障时刻相一致的状态。对于动态备份，装入数据库备份后，数据库并不处于一致性状态，还需要根据日志文件，利用系统故障恢复的方法，撤销备份结束时尚未完成事务对数据库的更新操作，并重做故障前已完成的事务，将数据库恢复到与故障时刻相一致的状态。

7.6.3　SQL Server 数据库备份与恢复技术

SQL Server 数据库中存在 4 种备份类型：

(1) 完整数据库备份；

(2) 差异数据库备份；

(3) 事务日志备份；

(4) 数据库文件或文件组备份。

完整数据库备份：任何其他数据库备份类型前，必须首先至少执行一次完整数据库备份。完整数据库备份是数据库恢复时的基线，执行完整数据库备份时，SQL Server 执行下列操作：备份在备份过程中发生的所有活动；备份事务日志中的所有未提交事务。

差异数据库备份：在执行差异备份之前必须已经执行了完整数据库备份。差异备份只备份自上一次完整数据库备份发生改变的内容和在差异备份过程中所发生的所有活动及事务日志中所有未提交的部分。差异数据库的恢复必须在完整数据库备份的基础上进行恢复。差异数据库备份的原理如图 7.10 所示。

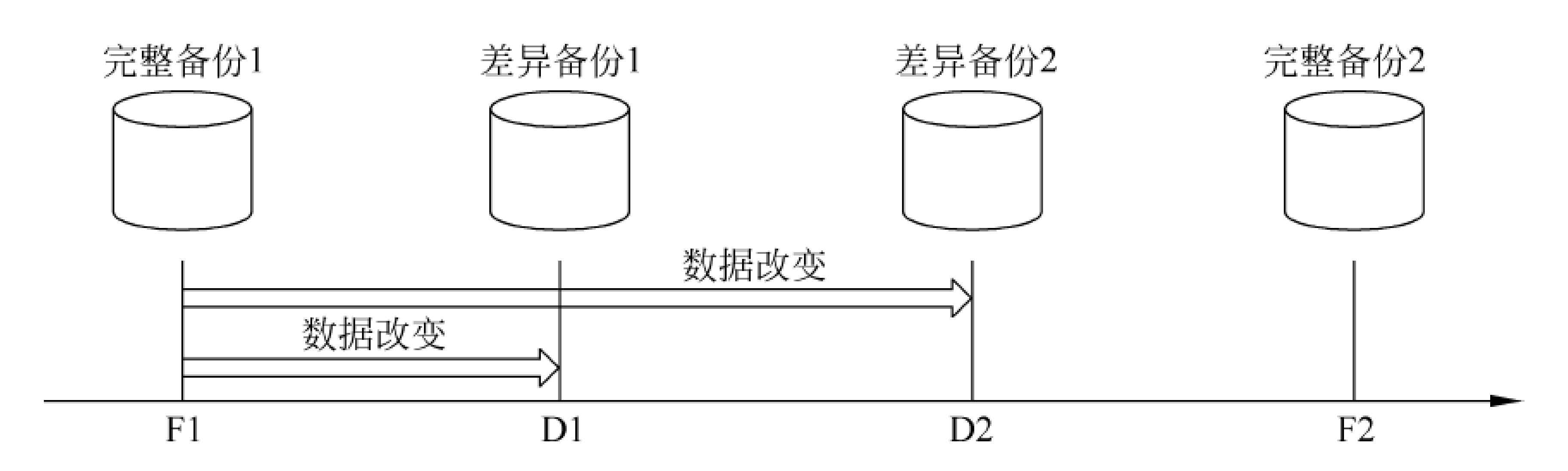


图 7.10　差异数据库备份原理图

事务日志备份：备份事务日志可以记录数据库的更改，但前提是在执行了完整数据库备份之后。进行事务日志备份时，SQL Server 执行备份操作是从上一次成功执行 BACKUP LOG 语句之后到当前事务日志结尾的这段事务日志，并从事务日志活动部分的起点处截断事务日志，丢弃不活动部分的信息。事务日志数据库备份的原理如图 7.11 所示。

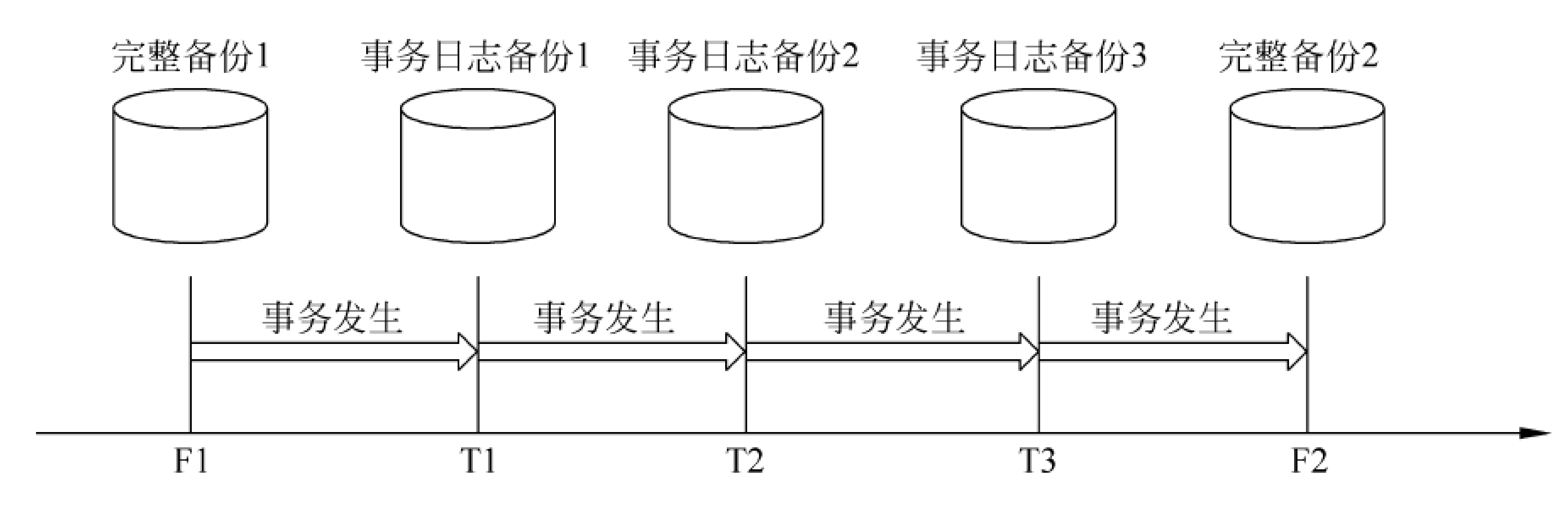


图 7.11　事务日志数据库备份原理图

数据库文件或文件组备份：对超大型数据库执行完整数据库备份是不可行的，可以执行数据库文件或文件组备份。必须指定逻辑文件或文件组，一般将表和索引一起备份。数据库文件或文件组的恢复通过完整数据库备份进行恢复，也可以单独恢复。

7.6.4　Oracle 数据库备份与恢复技术

1. Oracle 数据库备份的类型和方法

备份一个 Oracle 数据库有三种标准方式：导出（Export）备份、脱机备份（Offline Backup）和联机备份（Online Backup）。导出方式是数据库的逻辑备份，脱机备份和联机备份都是物理备份（也称为低级备份）。

1）逻辑备份

导出是将数据库中的数据备份到一个称为“导出转储文件”的二进制系统文件中。导出有以下三种模式。

（1）用户模式：导出用户所有对象及对象中的数据。

（2）表模式：导出用户的所有表或者用户指定的表。

（3）全局模式：导出数据库中的所有对象，包括所有数据、数据定义和用于重构数据库的存储对象。

导出备份可以导出整个数据库、指定用户或指定表。在导出期间，可以选择是否导出与表相关的数据字典的信息，如权限、索引和与其相关的约束条件，导出备份有以下三种类型。

（1）完全型：对所有表执行全数据库导出或仅对上次导出后修改过的表执行全数据库导出。

（2）积累型：备份上一次积累型备份所改变的数据。

（3）增量型：备份上一次备份后改变的数据。

导入是导出的逆过程，导入时读取导出创建的转储二进制文件以恢复数据。可以导入全部或部分已导出的数据。如果导入一个完全导出的整个导出转储文件，则所有数据库对象（包括表空间、数据文件和用户）都会在导入时创建。如果只打算从导出转储文件中导入部分数据，那么表空间、数据文件和将拥有并存储那些数据的用户必须在导入前设置好。

2）物理备份

物理备份是复制数据库文件而不是其逻辑内容。Oracle 支持脱机与联机两种物理备份。

（1）脱机备份。脱机备份在数据库已经正常关闭的情况下进行。数据库正常关闭后会提供给用户一个完整的数据库。当数据库处于脱机备份状态时，备份的文件包括所有数据文件、控制文件、联机重做日志和服务器参数文件。

当数据库关闭时，对所有这些文件进行备份可以提供一个数据库关闭时的完整镜像。以后可以从备份中获取整个文件集并恢复数据库的功能。

在磁盘空间容许的情况下，首先将需备份的文件复制到磁盘上，然后在空闲时将其备份到磁带上。脱机备份一般在 SQL * Plus 中进行。

例如，把 XSCJ 数据库的所有数据文件、重做日志文件和控制文件都备份。

① 正常关闭要备份的实例。

```
C:> sqlplus/nolog
SQL > connect system/manager as sysdba
SQL > shutdown normal
```

② 备份数据库。

使用操作系统的备份工具备份所有的数据文件、重做日志文件、控制文件和参数文件。

③ 启动数据库。

```
SQL > startup mount
```

(2) 联机备份。数据库可能要求24h运行,而且随时会对数据进行操作。联机备份可以在数据库打开的情况下进行,一般通过使用ALTER命令改变表空间的状态来开始进行备份,备份完成后恢复原来的状态,否则重做日志会错配,在下次启动数据库时引起表空间的修改。

进行联机备份时要求数据库必须在归档方式下操作,数据库不使用或使用率低,同时要有大量的存储空间。

数据库可从一个联机备份中完全恢复,并且可以通过归档的重做日志,前滚到任一时刻。只要数据库是打开的,当时在数据库中任一提交的事务都将被恢复,任何未提交的事务都将被回滚。联机备份的重要文件包括所有数据文件、归档的重做日志文件和一个控制文件。

Oracle以循环方式写联机重做日志文件,写满第1个日志后,开始写第2个,以此类推,当最后一个联机重做日志文件写满后,LGWR(Log Write)后台进程开始重新向第1个文件写入内容。当Oracle运行在ARCHIVELOG方式时,后台进程重写重做日志文件前将每个重做日志文件做一份复制件。

进行联机备份可以使用PL/SQL语句,也可使用备份向导。但都要求数据库运行在ARCHIVELOG方式下。

联机备份具备强有力的功能,其原因有两个:第一,提供了完全的时间点恢复;第二,在文件系统备份时允许数据库保持打开状态。

2. Oracle数据库恢复技术

最简单的恢复是使用最新的导出转储文件,使用Import命令,有选择地导入所需要的对象和用户。利用恢复向导进行恢复前,也需要和Oracle Management相连。若数据库处于打开状态,则只能恢复表空间或数据库文件。要恢复整个数据库,数据库必须处于装载状态。

Oracle数据库运行在ARCHIVELOG下使用恢复向导进行恢复的步骤如下:

(1) 数据库处于装载状态,单击【执行恢复】按钮,进入【执行恢复】界面,可以选择对整个数据库进行恢复或对某个对象进行恢复。选择【整个数据库恢复】类别中的【恢复到当前时间或过去的某个时间点】,在【主机身份证明】类别的用户名和口令文本框输入操作系统的用户名和对应的口令。

(2) 单击【执行整个数据库恢复】按钮,进入【时间点】界面,在此设置整个数据库恢复到当前时间还是以前某个时间点。

(3) 选中【恢复到当前时间】,单击【下一步】按钮,进入【重命名】界面,设置是否将文件

还原到其他位置。如果选择【是，将文件复制到新的公用位置】选项，那么将控制文件更新为使用新位置。

(4) 单击【下一步】按钮，进入【复查】界面。

(5) 单击【提交】按钮，完成恢复操作，剩下的工作由 Oracle 完成。恢复完成后，出现【恢复成功】界面。

数据库备份解决的主要问题是实例失败和磁盘失败。针对这两种失败类型，可以采用不同的恢复方法。

1) 实例失败

从实例失败中恢复应自动进行。数据库需要访问位于正确位置的所有控制文件、联机重做日志文件和数据文件。数据库中任何未提交的事务都要回滚。一个实例失败之后的那个数据库要重启动时，必须检查数据库报警日志中的错误信息。

当一个实例失败后数据库启动时，Oracle 检查数据文件和联机重做日志文件，并把所有文件同步到同一个时间点上，即使数据库未运行在 ARCHIVELOG 方式中，Oracle 也将执行这种同步。

2) 磁盘失败

磁盘失败也称为介质失败，通常由磁盘损坏或磁盘上的读错误引起，这样，一个磁盘上驻留的当前数据库文件将变得无法被数据库读出。驻留联机重做日志文件的磁盘应被镜像，失败时它们不会丢失。镜像可通过使用重做日志文件或操作系统级镜像文件实现。

如果丢失的是控制文件，不管选择什么备份方式都很容易恢复。每个数据库都有其控制文件的多个复制，且存储在不同的设备上。由 Oracle 安装程序生成的默认数据库创建脚本文件，只要关闭数据库创建三个控制文件，将它们放在三个不同的设备上。要恢复一个丢失的控制文件，只要关闭数据库并到保留有控制文件的地方复制一个到正确的位置即可。

如果所有控制文件都丢失，可以使用 CREATE CONTROLFILE 命令。该命令允许为数据库创建一个新的控制文件，并指定数据库中的所有数据文件、联机重做日志文件和数据库参数。如果对使用的参数有疑问并正在运行 ARECIVELOG 备份，可使用以下命令：

```
ALTER DATABASE BACKUP CONTROLFILE TRACE;
```

当执行这个命令时，一个合适的 CREATE CONTROLFILE 命令将写入到跟踪文件。这时可根据需要编辑由 Oracle 创建的跟踪文件。

如果丢失的是归档重做日志文件，就无法恢复。为此，最重要的是使归档制作日志文件，目标设备也保持镜像。归档重做日志文件与联机重做日志文件同等重要。

如果丢失的是数据文件，可从前一次的热备份中恢复，其步骤如下。

(1) 从备份中把丢失的文件恢复到其原来位置。

(2) 安装数据库

```
ORACLE_SID = CC1;     export ORACLE_SID
ORAENV_ASK = NO;     export ORAENV_AS
conect system/manager as sysdba
startup mount cc1;
```

(3) 恢复数据库,要求给出恢复所需的各归档日志文件名。

```
recover database;
```

出现提示时,为需要的归档重做日志文件输入文件名。另外,当数据库恢复重做发出提示时,可以使用 AUTO 选项。AUTO 选项使用定义的归档制作日志文件目标目录和文件名称格式,为归档重做日志文件名生成默认值。如果移动了归档重做日志文件,就不能使用该选项。

(4) 打开数据库

```
alter database open;
```

当从备份恢复数据文件时,数据库会辨认它是否来自数据库停止前的那个时间点。要找到那个时间点,就要应用归档重做日志文件里找到的事务。

7.7　数据库加密技术

随着对数据隐私和安全越来越多的关注和要求,信息技术提供者必须加强 DBMS 的安全策略和功能,以保护它们关键数据的安全。由于加密可以为 DBMS 提供更高的安全性,数据库加密技术在 DBMS 中被广泛采用。与一般的文件加密相比,对数据库中的数据实现加密有更大的难度,涉及到一般的文件加密不需考虑的许多问题。下面对数据库加密的要求及数据库加密的实现方法进行分析和阐述。

7.7.1　数据库加密要实现的目标

1. 传统的数据库加密技术

数据加密技术是一种重要的信息安全技术,它通过某种加密算法将明文数据变换成只有经过解密才可读的密文数据,使未经授权的访问即使得到密文数据也不能解读其信息。

传统的数据加密技术包括以下三种。

1) 对称加密

对称加密也称为共享密钥加密。发送者和接收者双方使用相同的密钥。由于安全强度高、加密速度快,成为最常用的加密技术。主要的加密算法有 DES 算法和 AES 算法。

2) 非对称加密

非对称加密又称为公钥加密。加密与解密双方使用不同的密钥,并且利用现有的技术和设备不能从公钥推出私钥。常用的公钥加密算法是 RSA,它不但可以用来加密数据,还可用来进行身份认证和数据完整性验证。

3) 混合加密

由于对称加密算法更简单,数据的加密和解密都使用同一个密钥,所以比起非对称加密,它的速度要快得多,适合大量数据的加密和解密;主要缺点也是由于使用相同的密钥加密和解密数据引起的,所有的数据发送方和接收方都必须知道或可以访问加密密钥,必须将

此加密密钥发送给所有要求访问加密数据的一方。所以在密钥的生成、分发、备份、重新生成和生命周期等方面常存在安全问题。而公钥加密属于非对称加密,不存在密钥的分发问题,因此在多用户和网络系统中密钥管理非常简单,但由于它主要基于一些难解的数学问题,所以安全强度没有对称加密高,速度也比较慢。

为了充分发挥对称加密与非对称加密的优势,混合加密方案被提出。在混合加密方案中,加密者首先利用一个随机生成的密钥和对称加密算法加密数据,然后通过使用接收者的公钥将随机密钥进行加密,并与密文一起传送给接收者。接收者通过自己的私钥首先解密随机密钥,再利用其解密密文。此方案既利用了对称加密安全强度高、速度快的特点,也利用了非对称加密密钥管理简单的特性。

"一次一密"的加密是最安全的一种加密技术,加密者在每次加密时都使用与明文长度一样的随机密钥,并且每个密钥都不重复使用。但在数据库加密中,由于密钥的产生和保存都存在很大的困难,因此在实际应用中并不常用。

2. 数据库加密实现的目标

与一般的数据加密和文件加密相比,由于数据库中的数据有很强的相关性,并且数据量大,因此对它的加密要比普通的数据加密和文件加密有更大的难度,密钥管理更加困难。目前,对数据库的加密方式主要分为软件加密和硬件加密。软件加密可以采用库外加密,也可以采用库内加密方式。库外加密方式即采用文件加密的方法,它把数据库作为一个文件,把每一个数据块当作文件的一个记录进行加密。文件系统与数据库管理系统交换的就是块号。库内加密按加密的粒度,可以进行记录加密,也可以进行字段加密,还可以对数据元素进行加密。数据元素加密时,每个元素被当作一个文件进行加密。硬件加密是在物理存储器(磁盘)与数据库文件之间加一个硬件装置,使之与实际的数据库脱离,加密时只对专一磁盘上的数据加密。

对数据加密的过程可将数据打乱,仅允许经过授权的人员访问和读取数据,从而确保数据的保密性,是一种有助于保护数据的机制。原始数据(称为"明文")与密钥值一起经过一个或多个数学公式处理后,数据就完成了加密。此过程使原始数据转为不可读形式。加密过的数据称为"密文"。为使此数据重新可读,数据接收方需要使用相反的数学过程以及正确的密钥将数据解密。

数据库加密要求做到:

(1) 数据库中信息保存时间比较长,不能采取一次一密的方法进行加密,应该选用其他加密方式,从实际上达到不可破译。

(2) 加密执行后,需要的存储空间不应明显增大。

(3) 加密/解密速度要快,尤其是解密速度要快,使用户感觉不到加密/解密时延和系统性能变化。

(4) 授权机制要尽可能灵活。在多用户环境中使用数据库系统,每个用户只用到其中一小部分数据。所以,系统应有比较强的访问控制机制,再加上灵活的授权机制配合起来对数据库数据进行保护。这样既增加了系统的安全性,又方便了用户的使用。

(5) 需要提供一套安全的、灵活的密钥管理机制。

(6) 不影响数据库系统的原有功能,保持数据库操作(如查询、检索、修改、更新)的灵活性和简便性。

（7）加密后仍允许用户对数据库的不同粒度进行访问。

7.7.2 数据库加密技术中的关键问题

数据库加密需要考虑几个重要问题：是在数据库引擎内或产生数据的应用程序中或是在硬件设备上进行加密/解密？密钥是存储在数据库中还是其他更安全的地方？加密数据粒度基于数据库、表还是字段？加密效果与其对性能的影响如何？

针对上述几个问题，结合数据库数据存储时间长、共享性高等特点，在数据库加密技术中，重点是要选择合适的加密执行层次、加密粒度和加密算法，并且要与实际的安全需求紧密结合起来。

1. 加密执行层次

数据库加密/解密的执行层次总体上分为两种：在 DBMS 内部执行和在 DBMS 外部执行，分别如图 7.12 和图 7.13 所示。

1）在 DBMS 内部执行加密解密

在 DBMS 内部执行加密/解密有如下特点。

（1）加密/解密执行时间：在数据存入数据库或从数据库中取出时，即在物理数据存取之前。

（2）加密解密执行主体：在 DBMS 内部，由用户定制的或者 DBMS 提供的存储过程/函数执行。

（3）加密/解密过程：在存储数据时，通过触发器调用加密存储过程对数据加密，然后将密文数据存入数据库；在读取数据时，触发器调用相应存储过程解密数据，然后读出结果。

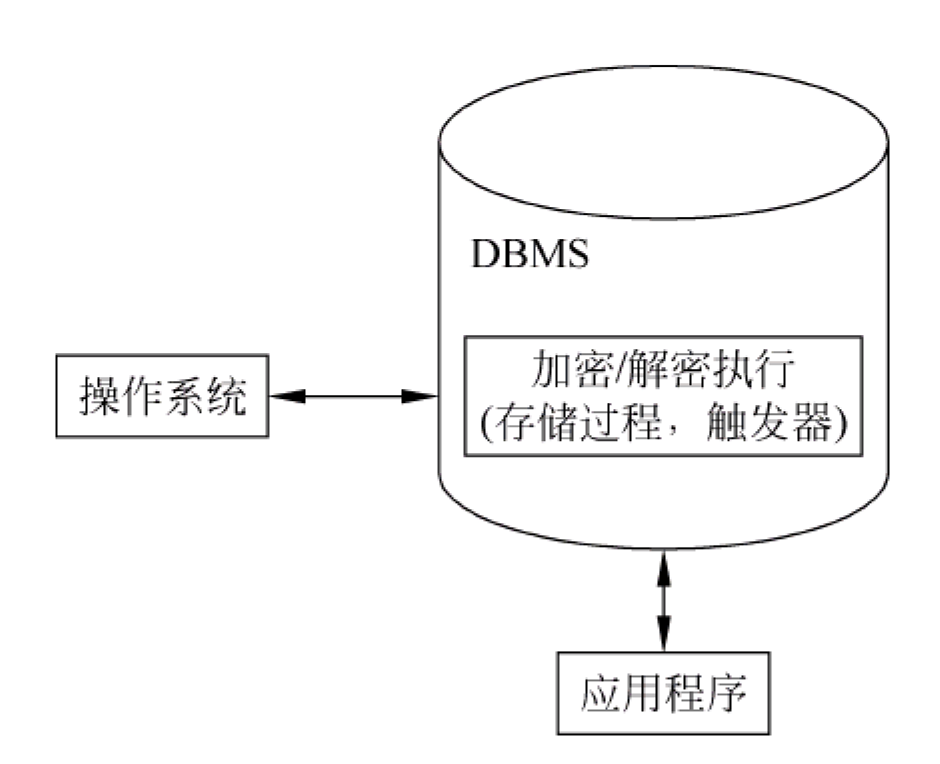


图 7.12 DBMS 内部实现数据的加密/解密

（4）加密/解密算法：由 DBMS 系统提供。多数 DBMS 不提供添加自己算法的接口，因此算法选择比较受限制。

在 DBMS 层执行加密/解密的方式，由于与 DBMS 系统结合紧密，可以提供对各种粒度加密的灵活性。灵活的加密配合 DBMS 的访问控制、授权控制，不失为一种有效的数据库数据保护方案。另外，这种层次的加密对于应用程序来说是透明的。因此，当安全需要升级的时候，这种方案下的应用程序基本上无须改动，从而安全升级非常方便。然而，这种层次的加密也存在着一些问题：加密需要数据库管理系统额外进行加密解密处理，包括每次数据存储/读取时，都会增加额外的处理负担，降低了系统运行的性能；密钥和密文没有分开存储，带来安全隐患；DBA 可以访问密钥表和加密数据表；可选择的算法受限制；为了解决性能和算法受限制的问题，也可以通过将加密功能嵌入 DBMS 内核或者在 DBMS 中引入一个加密子系统来完成加密，但相对太复杂，难以实现。

为了利用这种加密方式的优点，可以对密钥存储进行改进。单独存储密钥或采用二级密钥管理策略。单独存储密钥是将密钥存储在操作系统或者与数据库分离的独立硬件上，而采用二级密钥管理策略则可以加强密钥安全性。

2）在 DBMS 外部执行加密/解密

在 DBMS 外部执行加密/解密包含两种实现方式：第一种方式如图 7.13(a)所示，是在应用程序中实现，加密时调用应用程序中的加密模块来完成数据的加密工作，然后把密文数

据传送到DBMS中存储；解密时把密文数据取出到应用程序中，然后由应用程序中的解密模块将数据解密并给出结果。第二种方式如图7.13(b)所示，是直接利用操作系统提供的功能实现加密，这种加密方式是文件级别上的加密，直接加密数据库文件。

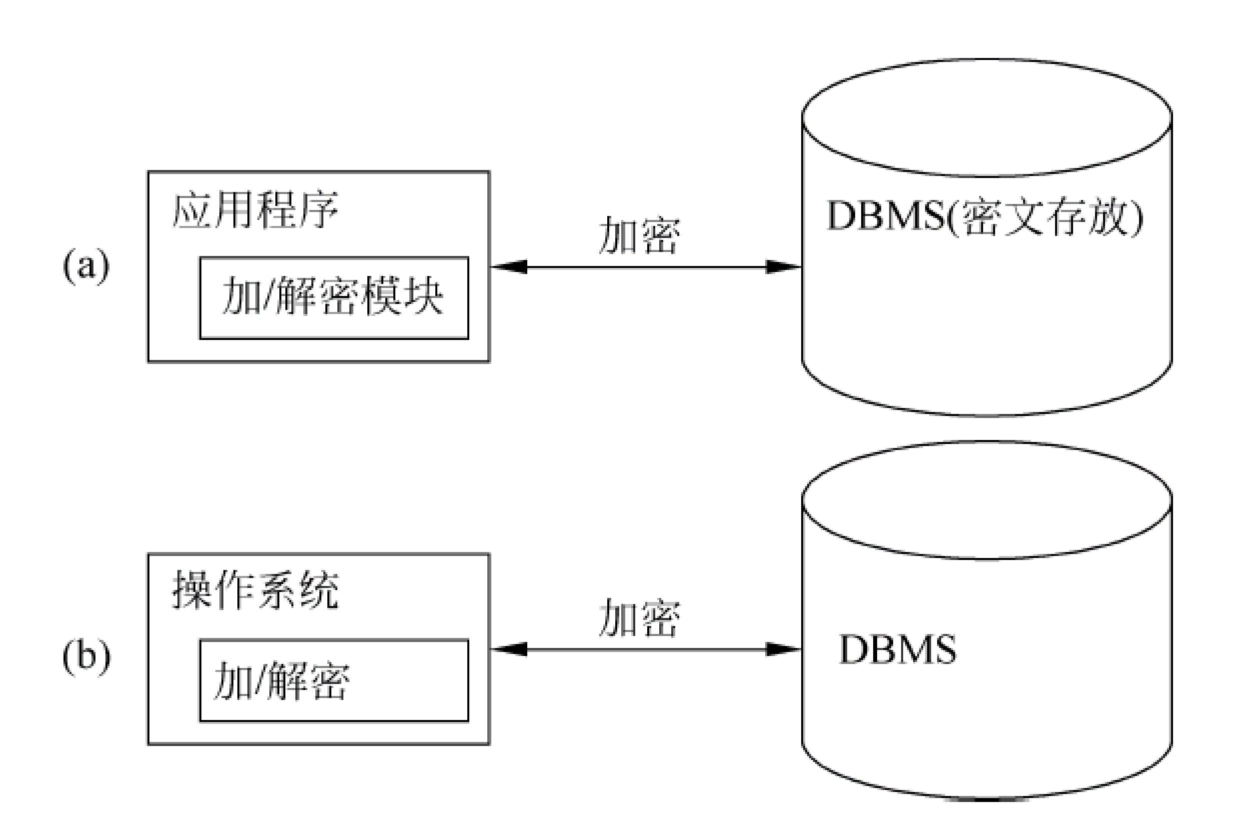


图7.13　DBMS外部执行数据的加密/解密

3）不同层次实现数据库加密效果比较

在操作系统(OS)层执行加密/解密，数据库元素以及各元素之间的关系无法辨认，所以无法产生合理的密钥。一般在OS层，针对数据库文件要么不加密，要么对整个数据库文件进行加密，加密/解密不能合理执行。尤其对于大型数据库来说，在操作系统层次实现数据库的加密/解密，目前还很难做到有效保证数据库的安全，因此一般不采用在OS层进行数据库加密解密。

在DBMS层执行加密/解密，不会增加额外的处理负担，因此对DBMS本身性能影响小；另外，实现了密钥与密文的分离，安全程度相对较高；算法由应用程序提供，选择性大，恶意DBA不知道算法，即使能够得到密钥和密文也较难得到明文信息。

直接在操作系统层对数据库进行加密，由于不能辨认数据库中数据的关系，因此数据加密的粒度非常粗糙，也无法产生合理的密钥，更谈不上密钥管理和保护。这种层次的加密在实际的数据库加密应用中，只能起到辅助作用，为数据库系统多添加一层保护。

DBMS外部加密主要存在着可用性与安全性的矛盾：加密粒度受DBMS接口支持的限制，灵活性不够强；安全升级时，应用程序改动比较大；对于密文数据，DBMS本身的一些功能会受到影响；DBMS内部扩展SQL语言中的一些内部函数将对加密数据失去作用。

DBMS管理系统中应用开发工具不能直接对加密数据进行操作，虽然有一些理论和方法可以实现对数据库密文进行操作，但如果能够在DBMS层次考虑加密问题，数据库加密将会有一个新的飞跃。

2. 加密算法选择

数据库加密技术的安全很大程度上取决于加密算法的强度，加密算法直接影响到数据库加密的安全和性能。因此，加密算法的选择在数据库加密方案中显得举足轻重。选择算法的主要指标是安全和便于使用。

安全的算法是可靠的算法。一个算法要可靠应该至少满足这样一些条件：首先，核心加密必须公认可靠，如DES和RSA算法，它们经过二十多年的考验、鉴定，数人攻击而无人承

认攻破；加密算法复杂度，主要是解密算法的复杂性要强，事实证明，仅靠对算法保密是不安全的，因为软件加密总可以通过反汇编可执行程序来发现；另外加密算法必须支持足够长的密钥，加密算法的实现必须可靠，这样才能具有较强的抗分析破译能力。

数据库加密中常用的对称密钥的分组密码算法有三重 DES、IDEA、RCS、RC6、Blowfish、CAST、RC2、AES。它们的明文按固定长度分组，对各组数据用不同的密钥加密（或解密）。这类密码按分组进行加密变换，若明文尾部不满一组，则用填充随机数的办法扩充其为整组再加密。该密文长度大于明文长度，但数据库加密后的数据长度不能改变，并且，分组加密算法中的 DES 算法密钥长度固定，无法扩展，并且已经被攻破，AES 算法至少 128 位分组，都不太适合数据库加密。

数据库中的数据特点之一是共享性，有权限的用户可能随时需要查询数据，因此，对加密速度有要求，要求加密后，数据量不能明显增加，数据长度不应改变，而且速度要尽量快。

根据数据库加密速度和安全性需要，加密数据采用对称加密算法。密钥管理则采用多级密钥管理体制（即分级加密管理模式）。

3. 加密粒度选择

数据库的加密粒度指的是数据加密的最小单位，主要有表、字段、数据元素等。在数据库中执行加密，加密粒度越小，则可以选择加密数据的灵活性就越大，但是产生的密钥数量也大，会带来管理方面的问题。数据库中加密粒度的选择要根据需要，充分衡量安全性和灵活性等需求。选择的过程中，由于数据库中存储的数据包括非敏感数据，因此，可以只选择敏感数据部分进行加密，从而加密粒度越小，加密执行消耗资源就少，投入费用就少。

加密粒度的选择可以是如下几种之一。

（1）属性值：可选的最小粒度，每个元组的属性值可以单独加密。

（2）记录/行：表中的每个记录单独加密。在知道需要加密的记录在表中的位置时可以不加密整个表内容，而选择记录/行作为最小粒度加密。

（3）属性/列：可以仅仅加密某些属性的列，如表中的信用卡号、身份证号、社会保障号等。

（4）页/块：一般自动加密处理时选择。需要加密的数据页/块存在硬盘上，对整个页/块进行加密，一个块包括数个元组（默认一般为 16KB）。

加密粒度的选择有一个原则：避免加密非敏感数据。如果一个记录只包含少数敏感字段，我们选择记录或块作为加密粒度，那么就会造成一定的浪费。而如果整个表都是需要加密的敏感数据，那么，选择页作为加密粒度效率会比较高。

以文件或列为单位进行加密，一方面密钥需要反复使用，加密系统的可靠性将会降低，另一方面加/解密一些不必要的信息可能会导致时间过长而无法使用。在目前的条件下，加/解密的粒度一般选择字段。

7.7.3 SQL Server 数据库加密技术

SQL Server 2000 以前的版本没有内置数据加密功能，若要在 SQL Server 2000 中进行数据加密，不得不买第三家产品，然后在服务器外部作 COM 调用或者是在数据送服务器之前在客户端的应用中执行加密。这意味着加密的密钥或证书不得不由加密者自己负责保护，而保护密钥是数据加密中最难的事，所以即使很多应用中数据已被很强地加密过，数据

保护仍然很弱。

SQL Server 2005 以后的版本通过将数据加密作为数据库的内在特性解决了这个问题。它除了提供多层次的密钥和丰富的加密算法外，最大的好处是用户可以选择数据库服务器管理密钥。

图 7.14 中服务器主密钥(Service Master Key)保护数据库主密钥(Database Master Keys)，而数据库主密钥又保护证书(Certificates)和非对称密钥(Asymmetric Keys)，最底层的对称密钥(Symmetric Keys)被证书、非对称密钥或其他的对称密钥保护，用户只需通过提供密码来保护这一系列的密钥。

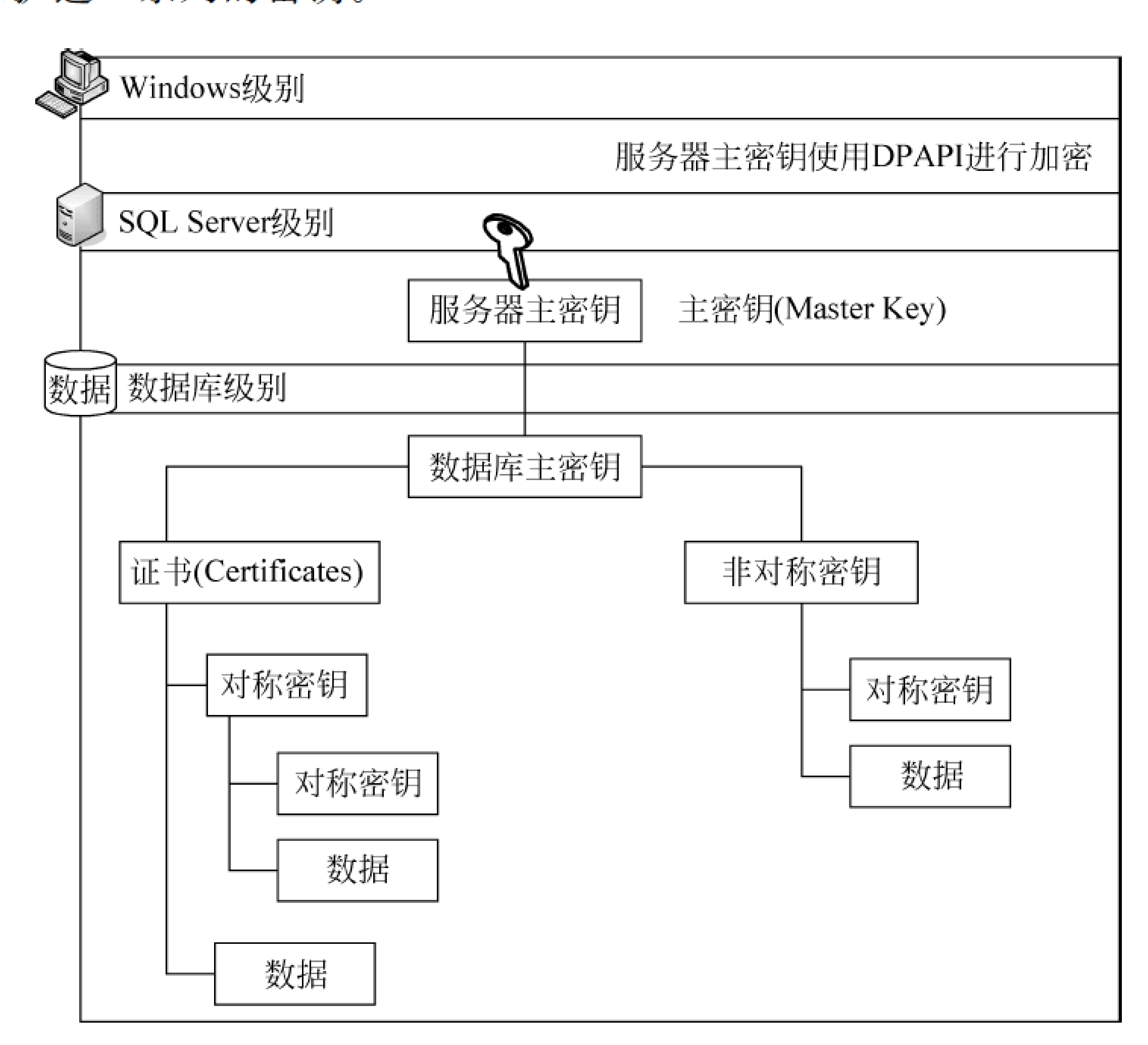


图 7.14 SQL Server 数据库密钥保护链原理图

1. 备份和恢复服务主密钥

由于服务主密钥是自动生成且由系统管理的，它只需要很少的管理，并且服务主密钥是 SQL Server 自动生成的，因此它没有对应的 CREATE 和 DROP 语句。服务主密钥可以通过 BACKUP SERVICE MASTER KEY 语句来备份，格式为：

```
BACKUP SERVICE MASTER KEY TO FILE = 'path_to_file'
ENCRYPTION BY PASSWORD = 'password'
```

参数 FILE='path_to_file'的意思是指定要将服务器主密钥导出到的文件的完整路径(包括文件名)。此路径可以是本地路径，也可以是网络位置的 UNC 路径。

参数 PASSWORD = 'password'的意思是用于对备份文件中的服务主密钥进行加密的密码，此密码应通过复杂性检查。

应当对服务器主密钥进行备份，并将其存储在另外一个单独的安全位置。创建该备份应该是首先在服务器中执行的管理操作之一。

【例 7-6】 备份服务主密钥。

```
BACKUP SERVICE MASTER KEY TO FILE = 'c:\service_master_key.txt'
ENCRYPTION BY PASSWORD = '123456';
```

如果需要从备份文件中恢复服务主密钥，使用 RESTORE SERVICE MASTER KEY 语句，基本语法格式为：

```
RESTORE SERVICE MASTER KEY FROM FILE = 'path_to_file'
DECRYPTION BY PASSWORD = 'password' [FORCE]
```

参数 FORCE 是指即使存在数据丢失的风险，也要强制替换服务器主密钥。

但是需要注意的是，如果在使用 RESTORE SERVICE MASTER KEY 时不得不使用 FORCE 选项，可能会遇到部分或全部加密数据丢失的情况。

【例 7-7】 恢复服务主密钥。

```
RESTORE SERVICE MASTER KEY FROM FILE = 'c:\service_master_key.txt'
DECRYPTION BY PASSWORD = '123456'
```

2. 创建、备份和恢复数据库主密钥

数据库主密钥通过 CREATE MASTER KEY 语句生成，其基本语法格式如下：

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'password'
```

备份数据库主密钥可以通过使用 BACKUP MASTER KEY 来进行，基本语法格式如下：

```
BACKUP MASTER KEY TO FILE = 'path_to_file'
ENCRYPTION BY PASSWORD = 'password'
```

恢复数据库主密钥使用 RESTORE MASTER KEY 语句，它需要使用 DECRYPTION BY PASSWORD 子句提供备份时指定的加密密码，还要使用 ENCRYPTION BY PASSWORD 子句来确认创建该数据库主密码时的密码，SQL Server 使用它提供的密码来加密数据库主密钥之后保存在数据库中。基本语法格式如下：

```
RESTORE MASTER KEY FROM FILE = 'path_to_file'
DECRYPTION BY PASSWORD = 'password'
ENCRYPTION BY PASSWORD = 'password'
[FORCE]
```

建议在创建了数据库主密钥之后立即备份数据库主密钥，并把它保存到一个安全的地方。同样，使用 FORCE 语句可能导致已加密数据的丢失。

3. 创建数字证书

SQL Server 数据库的数字证书可以使用 Management Studio 和 SQL 语句创建，下面是一个使用 SQL 语句创建数字证书的例子。

【例 7-8】 创建 SQL Server 的数字证书。

```
CREATE CERTIFICATE TestCertificate
ENCRYPTION BY PASSWORD = '123456'
```

```
WITH SUBJECT = 'This is a test certificate',
START_DATE = '12/28/2009',
EXPIRY_DATE = '1/1/2018';
```

例 7-8 创建了一个名为 TestCertificate、口令为 123456、有限期为 2009 年 12 月 28 日到 2018 年 1 月 1 日的数字证书。

4. 使用 SQL Server 数字证书加密/解密数据

通过内置的函数 EncryptByCert、DecryptByCert 和 Cert_ID，可以使用上面创建的数字证书来加密和解密数据。

1）Cert_ID 函数

Cert_ID 函数得到指定名字的证书的 ID。格式为：

```
Cert_ID ('cert_name')      -- cert_name 为证书的名字
```

2）EncryptByCert 函数

EncryptByCert 函数是指使用数字证书的公钥加密数据。只能使用相应的私钥对加密文本进行解密。此类非对称转换较比使用对称密钥进行加密和解密的方法，其开销更大。建议在处理大型数据集(如多个表中的用户数据)时不使用非对称加密。格式为：

```
EncryptByCert (certificate_ID,{ 'cleartext' | @cleartext })
```

该函数基本参数的含义如下。

(1) certificate_ID 为通过 Cert_ID 函数得到的证书 ID;

(2) cleartext 为要加密的明文，类型为 nvarchar、char、varchar、binary、varbinary 或 nchar;

(3) EncryptByCert 函数的返回值是最大大小为 8000 个字节的 varbinary。

下面，我们看看如何使用上述函数对数据库中的数据进行加解密。

【例 7-9】 对“某大学综合教务管理系统”teachsystem 的 cscore 表的 score 字段进行加密存储，SQL 语句如下：

```
insert into dbo.cscore values(1,1,1,EncryptByCert(Cert_ID('TestCertificate'),N'80'));
```

Score 字段加密后存储到数据库中的效果如图 7.15 所示，显示的是“二进制数据”。

使用普通的 SQL 查询语句无法查询加密数据的明文信息，必须使用 DecryptByCert 函数解密后再进行查询。

表 - dbo.cscore　localhost.teachsystem - SQLQuery5.sql

	sid	cid	score
▶	1	1	<二进制数据>
*	NULL	NULL	NULL

图 7.15　SQL Server 数据库加密效果图

【例 7-10】 查询“某大学综合教务管理系统”teachsystem 的 Tscores 表的加密数据，SQL 语句如下：

```
SELECT sid,cid,CONVERT(NVCHAR(50),
DECRYPTBYCERT(CERT_ID(N'TestCertificate'),
score,N'123456')) AS score FROM dbo.cscore
```

查询结果如图 7.16 所示，已经可以获得加密的 Score 字段的明文信息了。

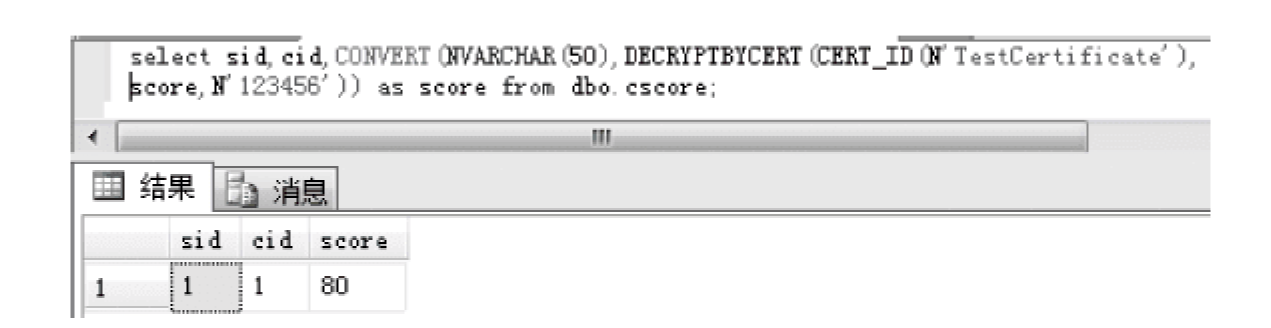


图 7.16 SQL Server 数据库解密效果图

7.7.4 Oracle 数据库加密技术

到目前为止，Oracle 提供两种加密方式：

(1) 加密 API，例如包 DBMS_CRYPTO(在 Oracle 10g 第 1 版和更高版本中)。使用这些包，我们可以构建自己的基础架构，对数据进行加密。这种方法的灵活性最强，但是构建和管理却相当复杂。

(2) 透明的数据加密是 Oracle 10g 第 2 版和更高版本的一个特性，使用该特性，我们就不必手动进行密码管理了。数据库管理密码，但是正如名称所指，加密是透明的——数据仅仅以加密的方式存储而已。当我们选择了这种方式，扩展性比较差。

为了更好地理解数据库的加密机制，我们将重点介绍第一种方法。在 Oracle 10g 中出现的 dbms_crypto 替代了之前的 dbms_obfuscation_toolkit，DBMS_CRYPTO 增加了若干新的加密算法、哈希算法。DBMS_CRYPTO 还撤销了对于 public 组的执行权限。对于 DBMS_CRYPTO 中的详细解释将后续分步介绍。

DBMS_CRYPTO 支持 DES 加密、双密钥的 3DES 以及三密钥的 3DES 加密，采用三个不同大小的 AES 和 RC4 加密算法。DBMS_CRYPTO 通过数字值如 DBMS_CRYPTO.encrypt_aes，这些参数是包的全局变量，可通过包名.变量，看起来有点像 Java 中 static final 类变量、C++ 中的 static constant，还有用数字表示的分组加密模式，如 CBC、CFB、ECB 和 OFB，以及填充模式 PKCS5、Zeros、ORCL 或 NONE。这些都作为常数变量传递给加密函数，Oracle 默认推荐 CBC 模式和 PKCS5。

DBMS_CRYPTO 包提供了一个加密函数和两个存储过程都取名为 encypt，但参数不同，这就是 PL/SQL 的函数和过程重载，需要的参数类型、参数数量和参数次序不同。函数基于 RAW 的数据类型，接收 RAW 数据类型的密钥、数据以及可选的初始向量(IV initialization vector)作为输入，并返回 RAW 数据类型。两道加密存储过程则是给予 LOB 数据类型，其中之一接收 BLOB 类型参数，而另外一道过程则接收 CLOB 类型参数，这两个存储过程都利用 RAW 数据类型的密钥和 IV，而且它们都是利用 BLOB 数据类型的 in out 参数。

下面详细介绍 Oracle 10g 以后的版本中 DBMS_CRYPTO 包的使用方法。

1. 简单随机数生成

使用 DBMS_CRYPTO 包可以有三个函数来生成简单的随机值，包括三种——数字、整数、字符。使用这些随机数生成函数是为了在加密时生成随机的密匙。

【例 7-11】 使用 DBMS_CRYPTO 包产生随机数。

```
SQL > SELECT DBMS_CRYPTO.RandomInteger FROM dual; -- 生成整数
SQL > SELECT DBMS_CRYPTO.RandomBytes(6) FROM dual; -- 生成 6 位 Bytes
```

```
SQL> SELECT DBMS_CRYPTO.RandomNumber FROM dual; -- 生成 Number
```

2. 使用 DBMS_CRYPTO 包进行数据加解密

为了方便地使用 DBMS_CRYPTO 包进行数据加解密，我们可以创建自己的数据加密函数 str_crypto 和解密函数 str_decrypt。

【例 7-12】 数据加密函数 str_crypto。

```
SQL> CREATE OR REPLACE FUNCTION str_crypto (string_in in varchar2) return raw is
string_in_raw RAW(128) := UTL_RAW.CAST_TO_RAW(string_in);
key_string varchar2(32) := 'test12345678901234567 8@163.com';
key_raw RAW(128) := UTL_RAW.CAST_TO_RAW(key_string);
encrypted_raw RAW(128);
BEGIN
  encrypted_raw := dbms_crypto.Encrypt(src => string_in_raw,
  typ => DBMS_CRYPTO.DES3_CBC_PKCS5,
  key => key_raw);
  RETURN encrypted_raw;
END;
/
```

【例 7-13】 数据解密函数 str_decrypt。

```
SQL> create or replace function t_to_back(raw_in in raw) return varchar2 is
string_out varchar2(50);
key_string varchar2(32) := 'test12345678901234567 8@163.com ';
key_raw RAW(128) := UTL_RAW.CAST_TO_RAW(key_string);
decrypted_raw RAW(128);
begin
  decrypted_raw := dbms_crypto.Decrypt(src => raw_in,
  typ => DBMS_CRYPTO.DES3_CBC_PKCS5,
  key => key_raw);
   string_out := UTL_RAW.cast_to_varchar2(decrypted_raw);
   return string_out;
end;
 /
```

下面看看如何使用上面创建的数据加密函数 str_crypto 和解密函数 str_decrypt 进行数据库数据加解密。

首先需要创建示例表 test，创建的 SQL 语句为：

```
SQL> CREATE TABEL test(a int primary key,psw varchar2(100));
```

使用加密函数 str_crypto 插入三行加密数据：

```
SQL> INSERT INTO test VALUES(1, str_crypto ('password'));
SQL> INSERT INTO test VALUES(1, str_crypto ('123456'));
SQL> INSERT INTO test VALUES(1, str_crypto ('hello,world'));
```

然后使用解密函数 str_decrypt 可以查询到明文数据：

```
SQL> select a, str_decrypt (psw) psw from test;
```

```
A PSW
1 password
2 123456
3 hello,world
```

如果直接使用 SELECT a,psw from test 查询,则查询出来的是乱码。

7.8 数据库高级安全技术

本章前面的内容讨论了数据库的基础安全性,对于高安全要求的数据库系统而言,仅有这样的基础安全性是不够的。下面以 Oracle 实现的虚拟专用数据库机制(VPD)和基于标签的安全机制(OLS)为代表,介绍数据库系统的高级安全技术和方法。VPD 机制提供对基于内容的访问控制和环境敏感的访问控制的支持,OLS 机制为多级安全数据库系统提供了一个实用的解决方案。

7.8.1 VPD 机制及其工作原理

1. VPD 机制

虚拟专用数据库(VPD)机制是一种细粒度的、基于内容的记录级访问控制机制。它的思想是动态地、透明地给数据库访问语句附加上合适的谓词,以便根据实际的安全需求限定用户所能访问到的数据记录,从而达到为特定的安全需求提供特定的数据库数据记录集的效果。

由于谓词透明地附加到数据库访问语句上,用户也许根本意识不到这一点,在他们看来,好像系统为他们提供了一个量身定制的专用数据库一样,如图 7.17 所示。

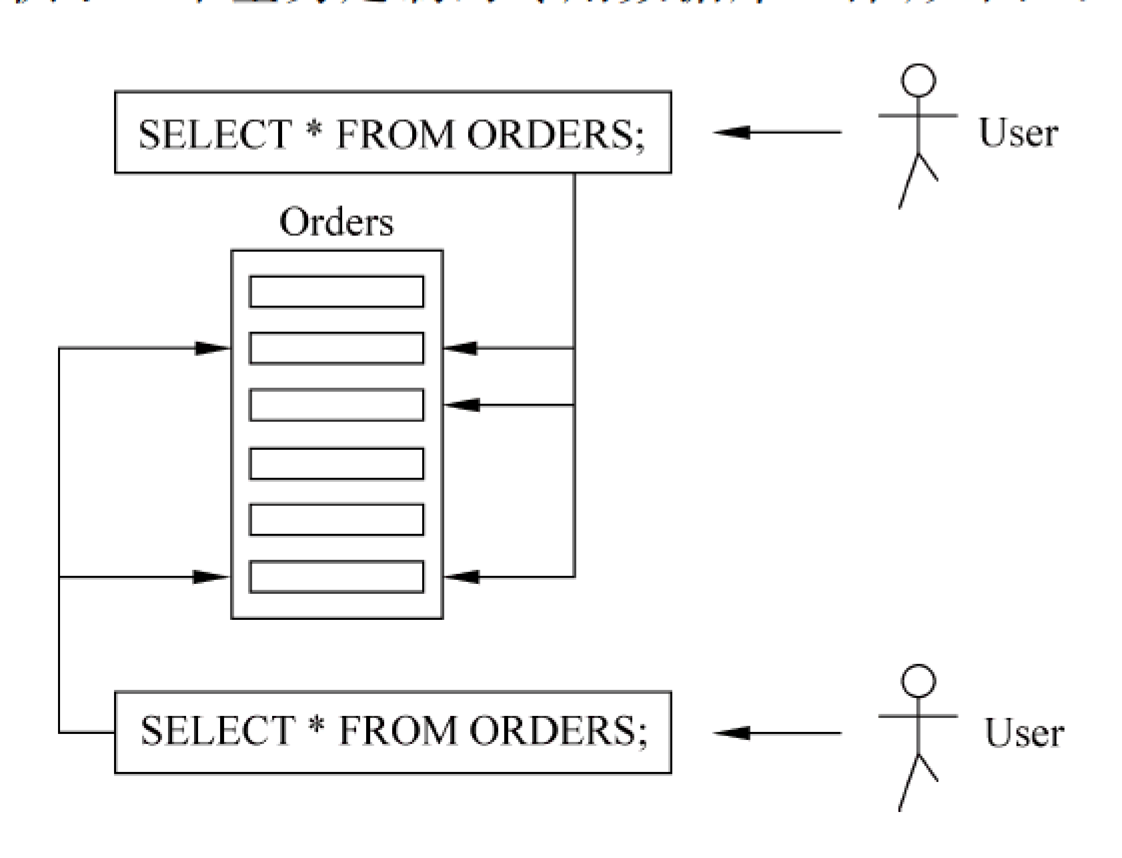


图 7.17 虚拟专用数据库示意图

在 Oracle 的 VPD 机制中,对数据库的访问控制方法由安全策略确定,一个安全策略的规则由一个函数来实现,每个函数的返回值都是一个字符串,这样的一个字符串就构成了一个谓词。一个函数定义之后,就可以与一张表关联起来,这样,每当要对这张表进行访问时,应用的谓词就自动附加到对应的访问语句上,从而在访问控制中发挥作用。

数据库记录的访问控制策略总是针对数据库表定义的。

【例 7-14】 对于保障信息表 MILIT. ASSUR 而言，一个简单的安全策略的例子是：不允许查询部门号为 10 的记录信息。下面的示例为这样的安全策略定义一个策略函数 no_dept10。

```
CREATE OR REPLACE FUNCTION no_dept10(
P_schema   IN VARCHAR2,
P_object   IN VARCHAR2)
RETURN VARCHAR2
AS
BEGIN
RETURN "deptno_10";
END;
```

如例 7-14 所示，每个策略函数都有两个参数：第一个参数是数据库模式的名称；第二个参数是数据库对象的名称，数据库对象可以是表或视图等。策略函数将与参数中给定的模式中的对象相关联。数据库系统在启动策略函数时，为策略函数的这两个参数提供参数值。

定义好的策略函数需要在系统中登记，为策略函数指定参数值，从而把该策略函数与一个指定模式中的指定的表关联起来。

【例 7-15】 上面定义的策略函数 no_dept10 可通过下面的操作与模式 MILIT 中的表 ASSUR 关联起来。

```
BEGIN
DBMS_RLS.add_policy
(object_schema      ⇒ "MILIT",
object_name         ⇒ "ASSUR",
policy_name         ⇒ "quickstart",
policy_function     ⇒ "no_dept10";
END;
```

此后，对 MILIT. ASSUR 进行访问时，函数 no_dept10 产生的谓词自动附加到访问语句上。如果用户执行以下操作，查询 ASSUR 表中的部门号：

```
SELECT DISTINCT deptno FROM emp;
```

则查询结果为：

```
DEPTNO
---------
20
30
```

显示结果中没有部门号 10，这是因为系统自动给其中的 SELECT 语句附加上了函数 no_dept10 产生的谓词，实际上，把 SELECT 语句改成了以下形式：

```
SELECT DISTINCT deptno FROM emp WHERE deptno!= 10
```

所以，执行 SELECT 语句时，部门号为 10 的记录被过滤掉了。当然，系统对 SELECT 语句的修改对于用户来说是透明的。从这个例子还可以看到，这里所说的谓词就是

WHERE 语句。

策略函数可以根据需要进行修改,不需要重新登记安全策略,函数修改后即可生效。

2. VPD 机制的工作原理

VPD 机制是绑定到表、视图等数据库对象上的访问控制机制,下面仅介绍针对数据库表的 VPD 机制。

VPD 机制根据安全策略实施访问控制,针对需要保护的数据库表,可以为它定义多种安全策略,每个安全策略由一个策略函数实现。每个策略函数具有两个参数,第一个参数是模式名,第二个参数是表名,通过这两个参数,一个策略函数可以和一个特定模式中的一张特定的表关联起来。每个策略函数产生一个字符串类型的返回值,这个返回值构成一个谓词。

当用户对受保护的表进行访问时,系统启动与该表关联的策略函数,策略函数产生相应的谓词,系统用该谓词构造一个 WHERE 子句,把它附加到原始的访问语句上,把原始的访问语句修改为新的访问语句。

策略函数的参数值是在对安全策略进行登记时确定的,安全策略的登记通过存储过程 DBMS_RLS. ADD_POLICY 来完成。在登记安全策略时,需要给出模式名、表名、安全策略名、策略函数名等信息。

1) 基于访问类型的访问控制

安全策略可以针对 SELECT、INSERT、UPDATE、DELETE 或 INDEX 等具体类型的语句实施。其中,针对 INDEX 类型语句的安全策略将影响 CREATE INDEX 和 ALTER INDEX 等命令的操作。

当用户对保护的表进行访问时,VPD 的访问控制机制就立即开始工作,检查访问语句的类型和相关安全策略,执行对应的策略函数,修改原始访问语句,并执行修改后的访问语句。图 7.18 是 VPD 机制的访问控制过程的一个示例。

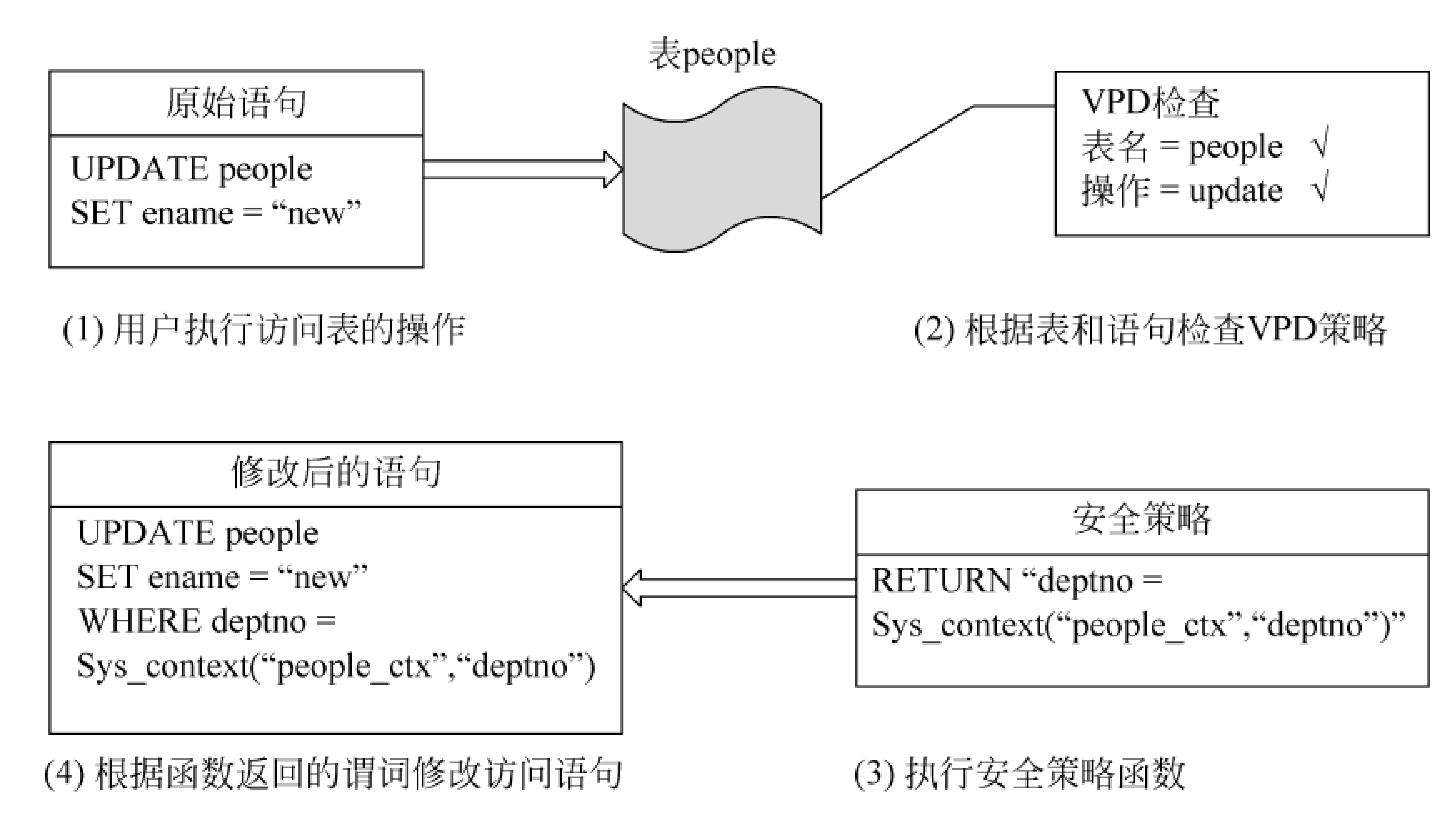


图 7.18 VPD 机制的访问控制过程

在默认情况下,一个安全策略将实施到所有类型的数据库访问语句中,对所有类型的访问语句都实施控制。

如果要使一个安全策略只针对 SELECT、INSERT、UPDATE、DELETE 或 INDEX 中的一类或几类语句实施控制，可以在登记安全策略时明确说明。如果在登记一个安全策略时，通过 STATEMENT_TYPES 指定了访问语句的类型，那么，该安全策略只对指定类型的语句起作用，这样可以实现基于访问类型的安全策略。

例如，可以制定一个安全策略，允许 SELECT 语句访问所有的记录；另外，制定一个 INSERT 和 UPDATE 型安全策略，使用户只能对他所在的部门的记录进行插入或更新操作；并且，制定一个 DELETE 型安全策略，使用户只能删除他自己的记录。

2）多个安全策略的访问控制

可以对同一张表实施多个安全策略，当多个安全策略实施到同一张表上时，系统对多个安全策略的控制实行逻辑“与”操作。有时，两个安全策略可能存在冲突，为了解决这样的冲突，可以在 VPD 机制中对安全策略实行分组管理，分组时确保同一个组中的安全策略不产生冲突结果。根据具体的应用环境，启用一些安全策略和禁用一些安全策略，同样可避免安全策略冲突情况的发生，可以通过在环境参数中配置安全策略的方法来达到这一目的。

每个会话都有相应的环境参数，把安全策略的名称设置到会话的环境参数中，可以确定在一个会话中要实施的将是哪个安全策略。在建立一个会话时，系统自动启用该会话的环境参数中设置的那些安全策略，从而提供基于会话的安全策略支持，解决安全策略冲突的问题。

3）VPD 机制的安全机理

VPD 机制的安全性来自谓词的记录过滤作用，不管数据库的访问请求是怎样发出的，也不管访问请求是哪个用户发出的，谓词都要对原始的查询结果进行过滤。这种记录过滤机制实现了独立于应用的、一致的记录安全性，即不管与数据库数据打交道的是什么样的应用。VPD 机制的安全性与受保护的数据紧密地结合在一起，它实施的记录过滤过程对发出数据库访问请求的应用是透明的，这种安全性具有不可旁路性。

记录过滤并不是 VPD 机制特有的特性，它是数据库系统中的一个基本功能，可以用多种方法来实现。在应用程序中实现记录过滤是一种常见的方法。与基于应用的记录过滤方法相比，VPD 机制提供的是一种紧贴数据库的底层控制方法，更有利于实现一致和持久的安全支持。

应用程序的变化是非常快的，因为用户的需求总是在不断发生变化，相比而言，一个好的数据库模式的生存期总是比一个应用程序的生存期要长，所以，数据库底层的安全模型对于实现整体的数据安全性具有更重大的意义。

VPD 机制实现面向数据的安全支持，提供数据安全与应用程序的相对独立性。在 VPD 机制的框架下，修改安全策略时，无须变更应用程序；修改应用程序时，也不必变更安全策略。

7.8.2 基于访问类型的控制实施

前面已介绍，VPD 机制可以根据 SELECT、INSERT、UPDATE、DELETE 或 INDEX 等语句类型进行有针对性的访问控制。但是，这样的访问控制是如何实施的呢？下面以 INSERT、UPDATE 和 DELETE 语句为代表，介绍 VPD 机制基于语句类型的访问控制的实施方法。

首先,假设有如下的安全需求:

(1) 允许用户查看所有的数据记录;

(2) 只允许用户插入和更新他所在部门的数据记录;

(3) 只允许用户删除他自己的数据记录。

为实现以上需求,需要知道哪个用户正在执行操作、该用户的用户名是什么、该用户所在部门的部门号是什么。可以通过环境参数来获取这些信息,不过,在这里,不关心通过环境参数来获取这些信息的具体方法。

1. INSERT/UPDATE 访问控制的实施

【例 7-16】 为了实现"只允许用户插入和更新他所在部门的数据记录"的安全需求,需要为 INSERT 和 UPDATE 操作建立一个安全策略,策略函数定义如下。

```
CREATE OR REPLACE FUNCTION dept_only(
p_schema     IN varchar2 DEFAULT   NULL,
p_objectIN VARCHAR2 DEFAULT NULL)
RUTURN VARCHER2
AS
BEGIN
RETURN "deptno = sys_context("people_ctx","deptno")";
END;
```

以上定义的策略函数 dept_only 是在安全管理员模式中建立的,这样可借助不同的模式实现策略函数与数据的分离,有利于实现对安全策略的保护。例 7-16 中策略函数的两个参数指定了默认值,都是 NULL,这便于在对参数进行赋值前就可以对函数进行测试。下面的示例可以查看策略函数 dept_only 返回的谓词字符串。

```
SELECT dept_only predicate FROM DULL;
```

查询结果是:

```
PREDICATE

----------------------------------
Deptno = sys_context("people_ctx", "deptno")
```

【例 7-17】 为 INSERT 和 UPDATE 操作创建一个安全策略,该安全策略创建后,只要用户在模式 MILIT 中的表 PEOPLE 上执行插入或更新操作,系统就调用模式 SEC_MGR 中的策略函数 DEPT_ONLY 实施访问控制。

```
BEGIN
DBMS_RLS.add_policy
(object_schema     ⇒ "MILIT",
object_name        ⇒ "PEOPLE",
policy_name        ⇒ "PEOPLE_IU",
function_schema    ⇒ "SEC_MGR",
policy_function    ⇒ "Dept_Only",
statement_types    ⇒ "INSERT,UPDATE",
update_check       ⇒ "TRUE"),
END;
```

创建了 INSERT/UPDATE 安全策略后，执行以下 UPDATE 操作时：

```
Update people
Set ename = "<NEW_VALUE>"
Where deptno = sys_context("people_ctx", "deptno")
```

系统将自动把它修改为如下的形式并执行：

```
Update people
set ename = "<NEW_VALUE>"
where deptno = sys_context("people_ctx","deptno")"
```

下面是一个执行 UPDATE 操作的例子。

【例 7-18】 用户针对如表 7.3 所示的用户部门信息表执行 UPDATE 操作。

表 7.3　用户部门信息表

USERNAME	DEPTNO
ALLEN	30
BLAKE	30
ADAM	20

设用户所在部门的部门号为 20，他执行以下操作时，将有一条记录被更新：

```
UPDATE people
SET username = "GRIZZLY"
WHERE username = "ADAMS"
```

但用户执行以下操作时，将没有记录被更新：

```
UPDATE people
SET ename = "Bozo"
WHERE ename = "BLAKE";
```

没有记录更新是因为系统自动把执行语句修改为：

```
UPDATE people
SET ername = "Bozo",
WHERE ename = "BLAKE",
AND deptno = sys_context("people_ctx", "deptno");
```

由于用户所在部门的部门号为 20，所以执行的等价语句是：

```
UPDATE people
SET ername = "Bozo",
WHERE ename = "BLAKE",
AND deptno = 20;
```

下面是执行 INSERT 操作的例子。

【例 7-19】 设用户 SCOTT 所在部门的部门号为 20，他执行以下操作时，将有一条记录被插入到表中：

```
INSERT INTO people
(username,job,salary,deptno)
VALUES('KNOX','Clerk','3000',20);
```

但用户执行以下操作时，将没有记录被插入到表中：

```
INSERT INTO people
```

```
(username,job,salary,deptno)
VALUES('ELLISON','CEO','90000',30);
```

这里有一点需要注意，在创建 INSERT/UPDATE 安全策略时，给 UPDATE_CHECK 选项设置了 TRUE 值，这样，当用户执行插入操作时，系统将把该安全策略实施到插入语句中。UPDATE_CHECK 选项的默认值是 FALSE，如果没有把它设置为 TRUE，系统将不对插入操作实施该安全策略。

2. DELETE 访问控制的实施

【例 7-20】 为了实现"只允许用户删除他个人的数据记录"的安全需求，需要为 DELETE 操作建立一个安全策略，下面的语句定义了所需的策略函数 user_only：

```
CREATE OR REPLACE FUNCTION user_only(
P_schema   IN  VARCHAR2 DEFAULT  NULL,
P_object   IN  VARCHAR2 DEFAULT  NULL),
RETURN VARCHAR2
AS
BEGIN
RETURN 'username = sys_context("userenv", "session_user")';
END;
```

下面的操作将创建一个 DELETE 安全策略：

```
BEGIN
DBMS_RLS.add_policy
(object_schema      ⇒ "SCOTT",
object_name         ⇒ "PEOPLE",
policy_name         ⇒ "PEOPLE_DEL",
function_schema     ⇒ "SEC_MGR",
policy_function     ⇒ "user_only",
statement_types     ⇒ "DELETE");
END;
```

创建了 DELETE 安全策略后，用户将只能删除用户名等于他自己的数据记录。如用户 SCOTT 执行 DELETE FROM people 操作，将删除且只删除用户名为 SCOTT 的数据记录。

7.8.3 VPD 安全防线

VPD 机制是数据库系统传统的安全防线之外的另一道安全防线，通过 VPD 机制可以增强系统的安全性，VPD 机制甚至可以防止拥有 DBA 角色的用户访问受保护的表。在创建了 DELETE 安全策略的情况下，拥有 DBA 角色的用户也无法删除表中的数据记录。

当然，虽然 VPD 机制的控制对 DBA 是有效的，但是，DBA 有权执行更改策略函数、删除数据库表、停用 VPD 安全策略等操作，所以，DBA 不会因恶意用户滥用 VPD 机制而永远失去对数据库表的访问能力。

除针对访问类型的控制，VPD 机制还可用于完全禁止用户访问数据记录，其方法是设计一个返回恒假谓词的策略函数。

【例 7-21】 返回字符串"1=0",下面定义的就是一个这样的函数。

```
CREATE OR REPLACE FUNCTION no_records(
p_schema     IN  VARCHAR2 DEFAULT  NULL,
p_object   IN  VARCHAR2  DEFAULT  NULL)
RETURN  VARCHAR2
AS
BEGIN
RETURN "1 = 0"
END;
```

假设用户 SCOTT 创建了表 people_ro,下面的操作将创建一个安全策略,禁止所有用户对表 people_ro 进行插入、更新或删除操作:

```
BEGIN
DBMS_RLS.add_policy
(object_schema      ⇒ "SCOTT",
object_name        ⇒ "PEOPLE_RO",
policy_name        ⇒ "PEOPLE_RO_IUD",
function_schema    ⇒ "SEC_MGR",
policy_function    ⇒ "NO_RECORDS",
statement_types    ⇒ "INSERT,UPDATE,DELETE",
update_check       ⇒ TRUE);
END;
```

在正常情况下,即使是 DBA 和数据的拥有者,也摆脱不了 VPD 机制安全策略的控制。有时,这会给数据库的维护工作造成障碍,如妨碍数据备份工作的进行。针对这样的问题,VPD 机制提供一个免受控制的权限,即 EXEMPT ACCESS POLICY 权限,获得该权限的用户,在访问数据库数据时,可以摆脱 VPD 机制所有策略函数的限制,就像根本没有任何策略函数存在一样。

EXEMPT ACCESS POLICY 是非常强大的权限,在授权时,需要特别谨慎。下面的语句可以查询系统中获得了该权限的用户的信息:

```
SELECT grantee
FROM dba_sys_privs
WHERE PRIVILEGE =  'EXEMPT ACCESS POLICY';
```

7.8.4　面向敏感字段的 VPD 功能

VPD 机制可以针对表的敏感字段启动访问控制功能,也就是说,可以把一张表中的某些字段确定为敏感字段,并告诉系统,当且仅当用户对表中的敏感字段进行访问时,才启动相应的 VPD 安全策略进行访问控制,如果用户只对表中的其他字段访问,则不必启动 VPD 安全策略的访问控制功能。

【例 7-22】 某些特殊军事单位设置的部门是敏感的,在单位人员信息表中,可以确定部门字段为敏感字段,利用 VPD 机制面向敏感字段的访问控制特性,可以允许任何用户查看人员信息表中除部门字段以外的所有信息,而只允许用户查看自己的部门信息。

下面的操作将创建一个面向 DEPT 敏感字段的 SELECT 策略：

```
BEGIN
DBMS_RLS.add_policy
(object_schema     => "SCOTT",
object_name        => "PEOPLE",
policy_name        => "people_sel_dep",
function_schema    => "SEC_MGR",
policy_function    => "user_only",
statement_types    => "SELECT",
sec_relevant_cols  => "DEPT");
END;
```

策略中的 SEC_RELEVANT_COLS 选项的值确定了 DEPT 字段为敏感字段，也确定了所创建的安全策略 people_sel_dep 为面向敏感字段的安全策略，当且仅当查询操作涉及 DEPT 字段时，该安全策略才实施控制作用。

当查询操作涉及敏感字段时，上面定义的安全策略只允许用户查询自己的记录信息。可以对该安全策略做一点修改，以便允许用户查询其他的记录信息，但屏蔽掉那些记录中的敏感字段信息。下面是修改过的安全策略的描述：

```
BEGIN
//移除现有安全策略
DBMS_RLS.add_policy
(object_schema         => "SCOTT",
object_name            => "PEOPLE",
policy_name            => "people_sel_dept");
//重新登记安全策略,增加 SEC_RELEVANT_COLS_OPT 选项
DBMS_RLS.add_policy
(object_schema         => "SCOTT",
object_name            => "PEOPLE",
policy_name            => "people_sel_dept",
function_schema        => "SEC_MGR",
policy_function        => "user_only",
statement_types        => "SELECT",
sec_relevant_cols      => "DEPT",
sec_relevant_cols_opt  =>DBMS_RLS.all_rows);
END;
```

修改后策略中的 SEC_RELEVANT_COLS_OPT 选项的值表示允许查询所有记录的数据，使那些在策略函数的限定下查询不到的记录也能够被查询出来，不过，那些记录的敏感字段的数据都将是不可见的。

面向字段敏感性的 VPD 功能对于支持私密性安全需求的实现具有重要意义，它允许把敏感信息和非敏感信息存放在一起，同时能够确保敏感信息不会泄露给无关用户。

7.9 数据库安全评估准则

身份认证、访问控制、安全审计、数据加密、备份恢复等安全机制可以在一定程度上增强数据库的安全性，实现了这些技术后，需要对数据库的安全性进行评估。评估一个数据库的

安全性，主要参考以下几个基本要求。

(1) 完全性：系统是否能对付各种可能的攻击。系统要达到什么样的完全性，取决于系统所处理信息的重要程度和敏感程度。

(2) 可信性：系统确实能对付各种可能的攻击的可信程度。可以通过对系统进行分析、证明、测试和渗透研究来获得系统的可信度。一般来说，要对系统进行证明是很困难的。

(3) 系统灵活性：系统要能实现各种不同的安全策略。如果系统中设计了某种安全策略，则系统的灵活性会比较差。因为不同的情况下，需要不同的安全策略，因此系统应有较大的灵活性。

(4) 便于使用：在安全功能和安全管理员之间的接口要简单。接口复杂比较容易出错，常会导致应授权的用户没有被授权、而不该授权的用户却被授权了这样的错误发生。

(5) 用户灵活性：系统不应给用户施加一些不必要的限制，如进行认证时的复杂协议、写应用程序时必须遵从严格的规则等。

(6) 防篡改：安全机制要能防止对它本身进行的非授权的修改。这一点是必须的，因为即使安全机制被证明是正确的，以后对它进行修改也会破坏其安全性。

(7) 开销小：由实现安全策略而带来的额外开销应比较小。

由于现阶段缺乏数据库安全评估工具，漏洞风险等级不断提升，数据库管理员无法及时发现数据库面临的安全问题。渗透测试完全模拟黑客的入侵和攻击手段，利用数据库存在的安全漏洞，在可控制和非破坏性的范围之内，对数据库进行模拟攻击，能够直观地向管理员反映数据库存在的安全漏洞。一般常用的数据库渗透测试工具有 DSQLTools(SQL 注入工具)、nbsi3.0(MSSQL 注入工具)、MySQLweak(MySQL 数据库弱口令扫描器)、pangolin(数据库注入工具)、db2utils(DB2 漏洞利用工具)、oscanner(Oracle 扫描工具)、Oracle_checkpwd_big(Oracle 弱口令猜解工具)等，但这些商用的漏洞扫描工具在实际应用中存在隐藏测试结果的问题，若无法对某种漏洞进行测试、无法了解存在漏洞的风险等级，数据库管理员会误以为数据库是安全的，可实际上却存在安全隐患。所以，提供具有可靠性和安全性的评估报告和漏洞风险等级是保证数据库管理员及时了解数据库是否安全的必要基础。

为保证数据库系统安全评估的过程更加完整、规范和公正，人们制定了数据库安全评估准则和保护轮廓等标准。国内外比较重要的数据库安全标准，包括根据美国国防部可信计算机系统评估准则 TCSEC 编制的指南性文件“TCSEC 在数据库管理系统中的解释(TDI)”、根据国际通用信息安全评估标准 ISO/IEC 15408(等同 CC 准则)制定的数据库管理系统的保护轮廓，以及我国出台的有关数据库安全的评估标准等。

在“橘皮书”发布之后，在美国国防部、国家计算机安全中心(NCSC)的支持下又对“橘皮书”进行了补充，出版了一些对“橘皮书”中的要求在各种特定环境下的解释，这些解释性的文件因为每本书使用不同颜色的书皮，所以人们又将之称为彩虹系列。其中，将 TCSEC 应用于数据库管理系统环境就形成了重要的 TDI。

TDI 作为 TCSEC 的系列解释之一，发表于 1991 年 4 月。它不是一个独立文件，必须结合 TCSEC 一同使用。实际上，TDI 真正的作用并非仅仅局限于数据库管理系统，它的主要思想也可以用于其他可信应用的解释，数据库管理系统作为贯穿整个 TDI 文档的一个典型的应用案例。

20 世纪 90 年代中期之前，TDI 作为可信产品评估的一个重要的适用标准，在美国是可

信数据库管理系统的评价基础。TDI 作为一种数据库安全评估标准，能够指导开发者在新开发的商用 DBMS 产品中提供安全特性，即 TCSEC 中所提出的可信要求；另一方面，TDI 作为一种可度量的评估，为处理涉密、分级和其他敏感信息的安全数据库系统提供了一种等级评估的方法。

以美国为首的西方六国发布 CC 通则后，即开始实施 CC 评估准则，并定于 2002 年底完成由原先的 TCSEC 评估向 CC 评估的过渡。实施 CC 评估，重要的是开发各类产品和系统的保护轮廓(PP)。

在数据库管理系统产品市场中屡占魁首的 Oracle 公司，自 CC 标准发布以来就积极地参与制定 DBMS 的 PP。Oracle 公司在 1988 年发表了两个数据库管理系统的 1.0 版本的保护轮廓，分别是政府数据库管理系统保护轮廓 G. DBMS. PP 和商用数据库管理系统保护轮廓 C. DBMS. PP，并且通过了 MAP 的 PP 评估，成为经过正式登记注册的保护轮廓。2000 年 3 月，Oracle 公司又更新了版本，并合并以前的两个类别，提出了通用数据库管理系统的 2.1 版本的保护轮廓 DBMS. PP。

CC 体系关于 PP 的开发有标准的规范，并发布了“PP 和 ST 的产生指南”。在该指南中，特别推荐一个样本目录。DBMS. PP 完全依照 CC 2.1 关于 PP 的规范开发而来，目标评估保证级是 EAL3，提供了数据库管理系统的安全要求集，兼容三种认证模式：OS 认证、DBMS 认证、混合认证模式。

由于数据库管理系统在我国的广泛应用，数据库系统中存放了大量的数据，安全问题尤其突出。我国对数据库管理系统的安全评估也开展了大量工作。在我国，军方最早涉足于安全数据库的设计开发，同时，也是军方提出了我国最早的数据库安全标准，即 2001 年的“军用数据库安全评估准则”。公安部为推行“等级保护计算机系统”，也为数据库安全做出了令人关注的工作，并于 2002 年发布了公安部行业标准“GAT/389-2002 计算机信息系统安全等级保护数据库管理系统技术要求”。这两个标准是我国目前关于数据库安全的直接的两个标准。

GA/T 389-2002 是公安部推行的“等级保护体系”的重要标准之一，该标准与“通用要求”(GA/T 390-2002 计算机信息系统安全等级保护通用技术要求)一起，作为 GB 17859-1999《计算机信息系统安全保护等级划分准则》(以下简称《准则》)技术要求在数据库应用方面的指南，详细说明了为实现《准则》所提出的安全要求应采取的具体数据库安全策略和安全机制，以及相应的安全功能和保证措施。GA/T 389-2002 针对数据库系统的特定情况，提出了分级的安全技术要求。

GA/T 389-2002 遵循的分级准则是 GB 17859-1999 的 5 级划分准则，同时在 5 级安全要求的基础上添加了许多 CC 的安全要求。GA/T 389-2002 中基本的 DBMS 安全技术要求包括身份鉴别、自主访问控制、标记、强制访问控制、客体重用、审计、数据完整性、隐蔽通道分析、可信路径、推理控制、TSF 保护、资源利用、TCB 访问控制、配置管理、分发和操作、开发、指导性文档、生命周期支持、测试、脆弱性评定、TCB 安全管理。

由于 GB 17859-1999 是基于 TCSEC 提出的，虽然在 GA/T 389-2002 中融入了新的 CC 中的安全要求组件，但是如何保证这样的安全要求集是否完备、能否通过统一的分级满足多元化的用户功能需求，以及如何保证评估体系的逻辑一致性，仍然是需要充分实践、检验和认识的问题。

7.10 小　结

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。数据库系统安全成为人们关注的焦点，任何系统的安全保护措施都不是完美无缺的，蓄意盗窃、破坏数据的人总是想方设法打破控制。而数据库系统面临的威胁可能是内部人员破坏和外部的非法入侵，如何保障数据库系统的安全更加具有挑战性。

本章对数据库的安全性进行总体介绍，包括数据库安全的定义、数据库安全研究的发展历史和研究现状等；详细介绍数据库系统的身份认证技术，包括数据库中身份认证的概念和必要性、身份认证技术在数据库系统中的应用方法等内容；数据库系统访问控制技术，包括关系型数据库的自主访问控制、强制访问控制和基于角色的访问控制技术，以及数据库新型访问控制技术；数据库系统安全审计技术，包括数据库安全审计的概念和必要性、数据库安全审计的实施方法；数据库系统的备份与恢复技术；数据库的存储和传输加密技术。最后以 Oracle 为例介绍数据库系统中的高级安全技术。

通过本章内容的学习，读者可以深入了解数据安全的发展历史、研究现状和面临的安全威胁，掌握身份认证、授权与访问控制、安全审计、备份与恢复、加密等常用的数据库安全防护技术。

习　题

一、填空题

1. 数据库安全的层次模型包括________、________、________和________。

2. 关系数据库管理系统主要采用________、________和________三种授权访问控制模型。

3. ________是收集和记录与系统安全有关活动的基础上，对其进行分析处理、评估审查，查找系统的安全隐患，对系统安全进行审核、稽查和计算，追查造成安全事故的原因，并做出进一步的处理。

4. 传统的数据库加密技术包括________、________和________。

5. 备份一个 Oracle 数据库有三种标准方式，分别是________、________和________。

6. 数据库加密/解密的执行层次总体上分为两种，分别是________和________。

7. SQL Server 采用的身份认证模式有________和________。

二、选择题

1. 数据库中用户对数据的任何访问必须经过授权，且这样的授权必须符合某种既定规则，这就是所谓的(　　)。

A. 身份认证　　B. 授权访问控制　　C. 安全审计　　D. 备份恢复

2. 一个数据库审计系统的简单模型包括两个部分：审计数据采集器和(　　)。

A. 审计数据记录器　　B. 审计数据保护器

C. 审计数据分析器　　　　D. 审计数据备份器

3. 备份和恢复是两个互相联系的概念,(　　)是将数据信息保存起来;而(　　)则是当意外事件发生或者有某种需求时,将已备份的数据信息还原到数据库系统中。

A. 存储　　B. 备份　　C. 冗余　　D. 恢复

4. ________是一种重要的信息安全技术,它通过某种加密算法将明文数据变换成只有经过解密才可读的密文数据,使未经授权的访问即使得到密文数据也不能解读其信息。

A. 数据库备份恢复　　　　B. 数据库安全审计

C. 数据库加密　　　　D. 数据库身份认证

三、简答题

1. 试述数据库身份认证的概念和常用的身份认证方法。
2. 试述数据库安全审计的流程。
3. Oracle 数据库备份的基本原则是什么?
4. 数据库安全机制有哪些?

第 8 章　信息系统安全评价标准和等级保护

随着网络和信息技术的发展，信息系统的安全性变得越来越重要。面对一个信息系统，用户的首要担忧是：这个系统安全吗？所以，计算机系统的提供者需要对产品的安全特性进行说明，而用户则需要验证这些安全特性的可靠性。然而，普通的用户难以对产品的安全性进行准确和充分的验证，难以判断系统提供者所提供的安全特性的有效性。因此，由独立安全专家对计算机安全进行第三方评价是非常必要的。国际上有多种对计算机信息系统的安全性进行评估的标准体系，这些评价标准能够准确地表达信息系统的安全性要求以及评价信息系统安全性的方法和准则，其内容和发展深刻反映了对信息安全问题的认识程度。了解其状况和发展对信息安全技术的研究十分重要，也是开发和评测各种信息安全技术的依据。同时，为了保障我国重要网络和系统的安全，确立了以等级保护制度为核心的信息安全保障体系。

本章 8.1 节介绍了信息安全评价标准的发展历史，8.2 节、8.3 节介绍了在国际上有影响力的两个信息安全评估标准 TCSEC 和 CC 标准，8.4 节重点介绍了我国的信息系统安全评估标准，8.5 节介绍了我国的信息安全等级保护制度，重点介绍了信息系统等级划分的依据、等级保护工作的主要环节。

8.1　信息安全评价标准的发展

世界各国对信息安全评价标准的研究可以追溯到 20 世纪 60 年代后期，1967 年美国国防部(DoD)发布了 Defense ScienceBoard Report，对当时计算机环境中的安全策略进行了分析；20 世纪 70 年代后期，开始对当时流行的操作系统进行安全方面的研究。1983 年美国国防部发布了“可信计算机系统评价标准(Trusted Computer System Evaluation Criteria，TCSEC)”，由于它使用了橘色书皮，所以通常称为橘皮书。TCSEC 是在 20 世纪 70 年代的基础理论研究成果 Bell&La Padula 模型的基础上提出的，其初衷是针对操作系统的安全性进行评估，后来美国国防部在国家计算机安全中心的主持下制定了一系列相关准则，例如，可信任数据库解释(Trusted Database Interpretation)和可信任网络解释(Trusted Network Interpretation)。由于每本书使用了不同颜色的书皮，人们将它们称为彩虹系列。在 1985 年，TCSEC 再次修改后发布，一直沿用至今。TCSEC 将信息安全等级分为 4 类，从低到高分别为 D、C、B、A，每类中又细分为多个等级。“可信”即可信赖，安全可靠，该标准使用户能够对计算机系统内敏感信息操作的可信程度做出评估，同时给计算机行业的制造商提供一种可循的指导规则，使其产品能够更好地满足敏感应用的安全需求。

TCSEC 最初只是军用标准，后来延至民用领域，它是计算机系统安全评估的第一个正

式标准,具有划时代的意义。TCSEC最主要的不足是其仅针对操作系统进行评估而且只考虑了保密性需求,但它极大地推动了国际计算机安全的评估研究,德国、英国等纷纷制定了各自的计算机系统评价标准。

但欧共体认为评估标准的多样性有违欧共体的一体化进程,也不利于各国对评估结果的互认,于是德国信息安全局在1990年发出号召,与英、法、荷四国一起迈开了联合制定评估标准的步伐,最终推出了"信息技术安全评估标准",简称ITSEC,又称欧洲白皮书。除了吸取TCSEC的成功经验外,ITSEC首次提出了信息安全的保密性、完整性、可用性的概念,他们的工作成为欧共体信息安全计划的基础,并对国际信息安全的研究实施带来了深刻的影响。ITSEC也定义了7个安全级别,即E6形式化验证、E5形式化分析、E4半形式化分析、E3数字化测试分析、E2数字化测试、E1功能测试、E0不能充分满足保证。

加拿大也在同期制定了"加拿大计算机产品评估准则"的第一版,称为CTCPEC,其第三版于1993年公布,CTCPEC吸取了ITSEC和TCSEC的长处,并将安全清晰地分为功能性要求和保证性要求两部分。

上述两种安全性测评准则不仅包含了对计算机操作系统的评估,还包含了现代信息网络系统所包含的通信网络和数据库方面的安全性评估准则。

美国政府在此期间并没有停止对评估准则的研究,于1993年公开发布了联邦准则的1.0版草案,简称FC。在FC中首次引入了"保护轮廓(PP)"的重要概念,每一保护轮廓都包括功能部分、开发保证部分和测评部分,其分级方式与TCSEC不同,充分吸取了ITSEC、CTCPEC的优点,供民用以及政府、商业使用。

总的来说,这一阶段的安全性评估准则不仅全面包含了现代信息网络系统的整体安全性,而且内容也有了很大的扩展,不再局限于安全功能要求,还增加了开发保证要求和评估(分析、测试)要求,但这些标准分散于各国,度量标准也不尽相同,客观上阻碍了信息安全保障的国际合作和交流,统一的安全评估准则呼之欲出。

为了能集中世界各国安全评估准则的优点,集合成单一的、能被广泛接受的信息技术评估准则,国际标准化组织在1990年就着手编写国际性评估准则,但由于任务庞大以及协调困难,该工作一度进展缓慢。直到1993年6月,在6国7方(英、加、法、德、荷、美国国家安全局以及国家标准技术研究所)的合作下,前述的几个评估准则终于走到了一起,形成了《信息技术安全通用评估准则》,简称CC。CC的0.9版于1994年问世,而1.0版则于1996年出版。1997年,有关方面提交了CC 2.0版的草案版,1998年正式发行,1999年发行了现在的CC 2.1版,后者于1999年12月被ISO批准为国际标准,编号ISO/IEC 15408(注:本文以下以CC指代ISO/IEC 15408),至此国际上统一度量安全性的评估准则宣告形成。CC吸收了各先进国家对现代信息系统安全的经验和知识,为信息安全的研究与应用带来了深刻影响。信息安全评价标准的发展历史如图8.1所示。

CC的评估等级共分7级,即EAL1到EAL7,分别为功能测试,结构测试,系统测试和检验,系统设计、测试和评审,半形式化设计和测试,半形式化验证的设计和测试,形式化验证的设计和测试。

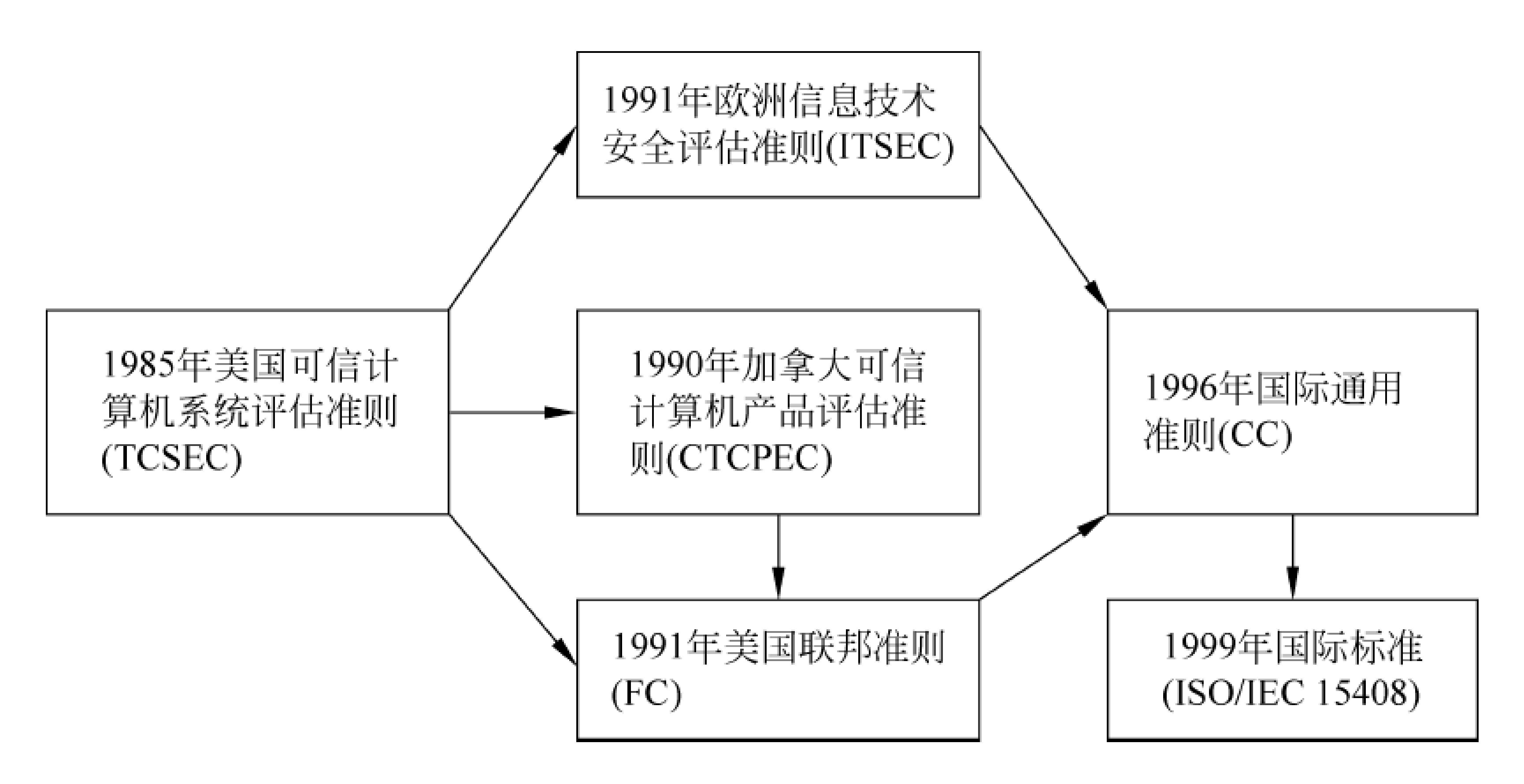


图 8.1　信息系统评估标准发展历史

8.2　可信计算机系统评价标准

8.2.1　TCSEC 的主要概念

在 TCSEC 中提出了以下主要概念，以描述计算机系统的安全问题。

(1) 安全性。包括安全策略、策略模型、安全服务和安全机制等内容，其中安全策略是为了实现软件系统的安全而制定的有关管理、保护和发布敏感信息的规定与实施细则；策略模型是指实施安全策略的模型；安全服务是指根据安全策略和安全模型提供的安全方面的服务；安全机制是实现安全服务的方法。

(2) 可信计算基(Trusted Computing Base，TCB)。TCB 是软件、硬件与固件的有机集合，它根据访问控制策略处理主体集合对客体集合的访问，TCB 中包含了所有与系统安全有关的功能。

(3) 主体(Subject)。计算机系统的主动访问者，如用户、代表用户运行的程序等。

(4) 客体(Object)。被访问或被使用的对象，如文件、内存、管道等。

(5) 自主访问控制(Discretionary Access Control，DAC)。DAC 是指资源的所有者(即属主)可以自主地确定其他用户对其资源的访问权。具有某类权限的主体能够将其对某资源(客体)的访问权直接或间接地按照需要动态地转让给其他主体或回收转让给其他主体。

(6) 强制访问权限(Mandatory Access Control，MAC)。MAC 是比自主访问控制更为严格的一种访问控制方式。在这种访问方式中，客体的访问权限不能由客体的拥有者确定，而是由系统管理者强制规定的。系统管理者为主体与客体规定安全属性(安全级别、权限等)，系统安全机制严格按照主体与客体的安全属性控制主体对客体的访问，对于系统管理员确定的安全属性，任何主体都不能修改和转让。

(7) 审计(Audit)。记录与系统安全相关的事件，以便对影响系统安全的活动进行追踪，确定责任者。

(8) 隐蔽信道。指一个进程利用违反系统安全的方式传输信息。有两类隐蔽信道，分

别是存储信道与时钟信道。存储信道是一个进程通过存储介质向另一个进程直接或间接传递信息的信道；时钟信道是指一个进程通过执行与时钟有关的操作把不能泄露的信息传递给另一个进程的通信信道，例如，一个文件的读写属性位可以成为隐蔽存储信道，而按某种频率创建与删除一个文件可以形成一个时钟隐蔽信道。

（9）客体重用：在计算机信息系统可信计算机 TCB 的空闲存储客体空间中，对客体初始指定、分配或再分配一个主体之前，撤销该客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时，当前主体不能获得原主体活动所产生的任何信息。

8.2.2 TCSEC 的安全等级

TCSEC 将可信计算机系统的评价规则划分为 4 类，即安全策略、可记账性、安全保证措施和文档。

安全策略包括自主访问控制、客体重用、标记、标记完整性、标记信息的扩散、主体敏感度标记、设备标记、强制访问控制等规则；可记账性包括标识与认证、可信路径、审计等规则；安全保证措施包括系统体系结构、系统完整性、隐蔽信道分析、可信设施管理、可信恢复、生命周期保证、安全测试、设计规范和验证、配置管理、可信分配等规则；文档包括安全特性用户指南、可信设施手册、测试文档、设计文档等规则。

根据计算机系统对上述各项指标的支持情况及安全性相近的特点，TCSEC 将系统划分为 4 类(Division)7 个等级，依次是 D、C(C1、C2)、B(B1、B2、B3)、A(A1)，按系统可靠或可信程度逐渐增高，如表 8.1 所示。

表 8.1 TCSEC 安全级别划分

等级分类	安全级别	定 义
A 类：验证保护类	A1	验证设计(Verified Design)
B 类：强制保护类	B3	安全域(Security Domain)
	B2	结构化保护(Structural Protection)
	B1	标记安全保护(Labeled Security Protection)
C 类：自主保护类	C2	受控的存取保护(Controlled Access Protection)
	C1	自主安全保护(Discretionary Security Protection)
D 类：最低保护类	D	最小保护(Minimal Protection)

在 TCSEC 中建立的安全级别之间具有一种偏序向下的兼容关系，即较高安全级别提供的安全保护要包含较低级别的所有保护要求，同时提供更多或更完善的保护能力。

1. D 最低保护类

D 级是最低保护级。将一切不符合更高标准的系统，通通归于 D 级，如 DOS 操作系统，它具有操作系统的基本功能，如文件系统、进程调度等，但在安全性方面几乎没有什么专门的机制来保障，是 D 级系统的典型例子。

2. C 自主保护类

该类的安全特征是系统的客体(如文件、目录等)可由其主体(如系统管理员、用户等)定义访问权限，自主保护类依据安全从低到高又分为 C1、C2 两个安全等级。

1）C1 自主安全保护级

只提供了非常初级的自主安全保护，可对用户按组实施授权，如 UNIX 系统中的 /owner/group/other 存取控制机制；能够实现对用户和数据的分离，进行自主访问控制(DAC)，保护或限制用户权限的传播。

C1 级系统是针对多个协作用户在同一敏感级别上处理数据的工作环境，其最主要的特点是把用户与数据隔离，提供自主访问控制功能，使用户可以对自己的资源自主地确定何时使用或不使用控制，以及允许哪些主体或组进行访问。通过用户拥有者的自主定义和控制，可以防止自己的数据被别的用户有意或无意地篡改、干涉或破坏。该安全级要求在进行任何活动之前，通过 TCB 去确认用户身份(如口令)，并保护确认数据，以免未经授权对确认数据的访问和修改。这类系统在硬件上必须提供某种程度的保护机制，使之不易受到损害；用户必须在系统注册建立账户并利用通行证让系统能够识别他们。C1 级要求较严格的测试，以检测该类系统是否实现了设计文档上说明的安全要求。另外还要进行攻击性测试，以保证不存在明显的漏洞让非法用户攻破而绕过系统的安全机制进入系统。另外，C1 级系统要求完善的文档资料。

2）C2 受控的存取保护级

C2 安全级具有以用户为单位的 DAC 机制，即将 C1 级的 DAC 进一步细化，保护粒度要达到单个用户和单个客体一级；C2 级增加了审计功能，审计粒度必须能够跟踪每个主体对每个客体的每一次访问，审计功能是 C2 较 C1 新增加的安全要求；C2 级还提供客体重用功能，即要求在一个进程运行结束后，要消除该过程残留在内存、外存和寄存器中的信息，在另一个用户过程运行之前必须清除或覆盖这些客体的残留信息；C2 级实施用户登录过程，对用户身份进行验证；C2 级实现资源隔离。C2 系统的 TCB 必须保存在特定区域中，以防止外部人员的篡改。

达到 C2 级的产品在其名称中往往不突出“安全”(Security)这一特色，很多商业产品已得到该级别的认证，如操作系统中 Microsoft 的 Windows NT 3.5、数字设备公司的 Open VMS VAX 6.0 和 6.1，数据库产品中 Oracle 公司的 Oracle 7、Sybase 公司的 SQL Server 11.0.6 等。

3. B 强制保护类

该类的安全特点在于由系统强制的安全保护，在强制保护模式中，每个系统对象(如文件、目录等资源)及主体(如系统管理员、用户、应用程序等)都有自己的安全标签，系统依据主体和对象的安全标签赋予它对访问对象的存取权限。强制保护类依据安全从低到高又分为 B1、B2、B3 三个安全等级。

1）B1 标记安全保护级

B1 在 C2 级的基础上增加了或加强了标记、强制访问控制、审计、可记账性和保障等功能，在 B1 级中标记起着重要的作用，是强制访问控制实施的依据。每个主体和存储客体有关的标记都要由 TCB 维护，不允许客体的拥有者改变其存取权限。

B1 级能够较好地满足大型企业或一般政府部门对数据的安全需求，这一级别的产品才能认为是真正意义上的安全产品。满足此级别的产品前一般多冠以“安全”或“可信”字样，作为区别于普通产品的安全产品出售。例如，操作系统方面，典型的有数字设备公司的 SEVMS VAX 6.0、惠普公司的 HP-UX BLS release 9.0.9+；数据库方面则有 Oracle 公司

的 Trusted Oracle 7、Sybase 公司的 Secure SQL Server 11.0.6、Informix 公司的 Incorported INFORMIX-Online/Secure 5.0 等。

2）B2 结构化保护级

B2 系统的设计中把系统内部结构化地划分成明确而大体上独立的模块，并采用最小特权原则进行管理。B2 级不仅要求对所有对象加标记，而且要求给设备（磁盘或终端）分配一个或多个安全级别（实现设备标记）。必须对所有的主体和客体（包括设备）实施强制性访问控制保护，必须要有专职人员负责实施访问控制策略，其他用户无权管理。通过建立形式化的安全策略模型并对系统内的所有主体和客体实施自主访问控制和强制访问控制。

B2 级较 B1 级有一项更强的设计要求，B2 级系统的设计与实现必须经得起更彻底的测试和审查，必须给出可验证的顶级设计（Top-Level Design），并且通过测试确保该系统能够实现这一设计。还需要对隐蔽信道进行分析，确保系统不存在各种安全漏洞。实现中必须为安全系统自身的执行维护一个保护域，必须确保该域的安全性不受外界的破坏，进而保护整个系统的目标代码和数据的完整性不受外界破坏。

目前，经过认证的 B2 级以上的安全系统非常稀少。例如，符合 B2 级的操作系统只有 Trusted Information Systems 公司的 Trusted XENIX 一种产品，符合 B2 标准的网络产品只有 Cryptex Secure Communications 公司的 LLC VSLAN 一种产品，而数据库方面则没有符合 B2 标准的产品。

3）B3 安全域保护级

B3 级的 TCB 必须满足访问监控器的要求，审计跟踪能力更强，并提供系统恢复过程。B3 安全级要求系统有主体/客体的区域，有能力实现对每个目标的访问控制，使每次访问都受到检查。用户程序或操作被限定在某个安全域内，安全域间的访问受到严格控制。这类系统通常采用硬件设施来加强安全域的安全，例如内存管理硬件用于保护安全域免受无权主体的访问或防止其他域的主体的修改。该级别要求用户的终端必须通过可信的信道连接在系统上。

为了能够确实进行广泛而可信的测试，B3 级系统的安全功能应该是短小精悍的。为了便于理解与实现，系统的高级设计（High Level Design）必须是简明而完善的，必须组合使用有效的分层、抽象和信息隐蔽等原则。所实现的安全功能必须是高度防突破的，系统的审计功能能够区分何时能避免一种破坏安全的活动。为了使系统具备恢复能力，B3 级系统增加了一个安全策略。

(1) 安全策略：采用访问控制列表进行控制，允许用户指定和控制对客体的共享，也可以命名指定用户对客体的访问方式。

(2) 可记账性：系统能够监视安全审计事件的发生与积累，当超出某个安全阈值时，能够立刻报警，通知安全管理人员进行处理。

(3) 保障措施：只能完成与安全有关的管理功能，对其他完成非安全功能的操作要严格限制。当系统出现故障与灾难性事件后，要提供一种过程与机制，保证在不损坏保护的条件下，使系统得到恢复。

4. A 验证保护类

A1 安全级又称为可验证性设计保护级，是橘皮书中最高的安全级别，它的安全功能要

求与 B3 一致，但是它包含了一个严格的设计、控制和验证过程，即提供 B3 级保护的同时给出系统的形式化设计说明和验证以确信各安全保护真正实现。

A1 安全级的设计要求非常严格，达到这种要求的系统很少。目前已获得承认的此类系统有 Honeywell 公司的 SCOMP 系统。A1 安全级标准是安全信息系统的最高安全级别，一般信息系统很难达到这样的安全能力。

8.2.3　TCSEC 的不足

TCSEC 是第一代的安全评估标准，它有自身的不足，但这并不意味着我们可以完全抛弃它。目前不止我们国内，即使在世界上也都存在着对 TCSEC 与 CC 优劣的争论，有很多还未达成一致性的意见。以下是当前已得到公认的对 TCSEC 局限性的认识。

(1) TCSEC 是针对建立无漏洞和非侵入系统制定的分级标准。TCSEC 不是基于时间的安全模型，而是基于功能、角色、规则等空间与功能概念意义上的安全模型。安全概念仅仅是为了防护，对防护的安全功能如何检查以及检查出的安全漏洞又如何弥补和反应等问题没有讨论和研究。

(2) TCSEC 是针对单一计算机，特别是针对小型计算机和主机结构的大型计算机制定的测评标准。TCSEC 的网络解释目前缺少成功实践支持，尤其对于互联网络和商用网络很少有成功的实例支持。

(3) TCSEC 主要用于军事和政府信息系统，对于个人和商用系统，采用这个方案是有困难的。也就是说其安全性主要是针对保密性而制定的，而对完整性和可用性研究的不够，忽略了不同行业的计算机应用的安全性的差别。

(4) 安全的本质之一是管理，而 TCSEC 缺少对保障(保证)的讨论。

(5) TCSEC 的安全策略也是固定的，缺少安全威胁的针对性，其安全策略不能针对不同的安全威胁实施相应的组合。

(6) TCSEC 的安全概念脱离了对 IT 和非 IT 环境的讨论，如果不能把安全功能与安全环境相结合，那么安全建设就是抽象的和非实际的。

(7) 美国 NSA 测评一个安全操作系统需要花一、二年以上的时间，这个时间已经超过目前一代信息技术发展的时间，也就是说 TCSEC 测评的可操作性较差，缺少测评方法框架和具体标准的支持。

8.3　通用评估标准

美、加、英、法、德、荷等国家联合推出的“信息技术安全性通用评估准则”(Common Criteria for Information Technology Security Evaluation，CC)于 1999 年 7 月通过国际标准化组织认可，确立为信息安全评价国际标准，其标准编号为 ISO/IEC 15408。CC 标准提出了“保护轮廓”，将评估过程分为“功能”和“保证”两部分，是目前系统安全认证方面最全面也是最权威的标准。CC 标准确立后，美国不再受理以橙皮书为尺度的新的评价申请，之后的安全产品评价工作均按 CC 标准进行。

8.3.1 CC的组成

CC分为三个部分,其中第一部分“简介和一般模型”介绍了CC中的有关术语、基本概念和一般模型以及与评估有关的一些框架,附录部分主要介绍了保护轮廓(PP)和安全目标(ST)的基本内容。

第二部分“安全功能要求”按“类-族-组件”的方式提出安全功能要求,提供了表示评估对象(Target Of Evaluation,TOE)安全功能要求的标准方法。TOE是指被评估的信息技术产品、系统或子系统,如防火墙、计算机网络系统、密码模块,以及相关的用户指南和设计方案等。除正文以外,每一个类还有对应的提示性附录作进一步的解释。

在CC标准中,安全要求(包括安全功能要求和安全保证要求)均以“类-族-组件”的形式进行定义。首先,对安全需求的全集,根据不同的侧重点,划分成若干大组,每个大组就称为一个类。每个类的安全需求,根据不同的安全目标,又划分成若干小组,每个小组就称为一个族。每个族的安全需求,根据不同的安全强度或能力,再进一步划分成更小的组,每一个这样的更小的组用一个组件来表示。这样,安全需求由类构成,类由族构成,族由组件构成。组件是CC标准中最小的可选安全需求集,是安全需求的具体表现形式。

例如,身份识别和认证方面的需求归为一个类,这个类中,身份识别方面的需求归为一个族;这个族中,缓时识别方面的需求构成一个组件;所谓缓时识别是指允许用户在身份识别前执行适当的操作。

CC标准定义了11个公认的安全功能需求类,即安全审计类、通信类、密码支持类、用户数据保护类、身份识别与认证类、安全管理类、隐私类、安全功能件保护类、资源使用类、评估目标访问类和可信路径/通道类。

安全审计类涉及与安全有关的操作信息的识别、记录、存储和分析等方面的需求;通信类涉及数据交换双方的身份确保等方面的需求,包括收、发双方的防抵赖等;密码支持类涉及密钥管理和加密操作等方面的需求;用户数据保护类涉及对用户数据进行保护的安全功能和安全策略等方面的需求;身份识别与认证类涉及证实用户身份和确立安全属性等方面的需求;安全管理类涉及对产品的安全功能件中的属性、数据和功能等进行管理方面的需求;隐私类涉及确保用户身份的隐蔽性和防止用户身份被盗用等方面的需求;安全功能件保护类涉及确保安全功能件中的有关机制和数据的完整性等方面的需求;资源使用类涉及对需要访问的资源的可用性给以支持等方面的需求;评估目标访问类涉及对用户和安全产品会话过程的建立进行控制等方面的需求;可信路径/通道类涉及在用户与安全功能件之间建立可信通信路径、在安全功能件与其他可信IT产品之间建立可信通信通道等方面的需求。

第三部分“安全保证要求”定义了评估保证级别,建立了一系列安全保证组件作为表示TOE保证要求的标准方法。第三部分列出了一系列保证组件、族和类,还定义了PP和ST的评估准则,并提出了评估保证级别。

CC标准定义了7个公认的安全保证需求类,即构造管理类、发行与使用类、开发类、指南文档类、生命周期支持类、测试类和脆弱性评估类。

构造管理类涉及确保产品的功能需求和规格说明在最终的安全产品中得以实现方面的需求。发行与使用类涉及安全产品的正确发行、安装、生成和投入运行等方面的需求。开发类涉及三方面的需求：一是安全功能件在不同抽象层次上的表示，二是不同抽象层次上的安全功能件表示之间的一致性，三是安全策略模型的建立及安全策略、安全策略模型与功能描述之间的一致性。指南文档类涉及产品的用户指南、管理员指南等文档资料方面的需求。生命周期支持类涉及产品的开发、维护过程中有关开发、维护模式以及安全措施等方面的需求。测试类涉及产品测试方面的需求。脆弱性评估类涉及对产品可能存在的脆弱性（如隐通道等）进行分析等方面的需求。

CC 的三个部分相互依存，缺一不可。其中第一部分介绍 CC 的基本概念和基本原理，第二部分提出了技术要求，第三部分提出了非技术要求和对开发过程、工程过程的要求。这三部分的有机结合具体体现在 PP 和 ST 中，PP 和 ST 的概念和原理在第一部分介绍，PP 和 ST 中的安全功能要求和安全保证要求在第二、第三部分选取，这些安全要求的完备性和一致性由第二、第三两个部分来保证。

8.3.2　需求定义的用法

安全需求定义中的类和族反映的是分类方法，具体的安全需求由组件体现，选择一个需求组件等同于选择一项安全需求。CC 标准鼓励人们尽可能选用该标准中已定义的安全需求组件，也允许人们自行定义其他必要的安全需求组件。每个安全需求组件表示的是某项具体的安全需求。通常，一个安全产品总是融多项安全需求于一身，需要用多个需求组件以一定的组织方式组合起来进行表示。CC 标准定义了三种类型的用于描述产品安全需求的组织结构，即“安全组件包”、“保护轮廓定义书（PP）”和“安全对象定义书（ST）”，安全需求组件可以在这三种类型的组织结构中得到使用。

1. 安全组件包

把多个安全需求组件组合在一起所得到的结果就叫做一个安全组件包。安全组件包可用于构造更大的安全组件包或用于构造 PP 和 ST。安全组件包可以表示一组安全功能需求或安全保证需求，这些需求可以满足预定安全目标中的某个子目标的需要。

2. 保护轮廓定义书（PP）

保护轮廓定义书是一份安全需求说明书，CC 标准对它的格式有明确的规定。PP 针对某一类安全环境确立相应的安全目标，进而定义为实现这些安全目标所需要的安全需求。PP 给出的是一个与实现无关的安全需求定义，它所定义的这些需求没有针对具体的某一种安全产品，只针对比较明确的安全目标。通常，同一个 PP 中所定义的安全需求可以在多种不同的安全产品中实现。每个 PP 都必须指定一个安全可信度级别，这是按照该 PP 研制的安全产品所应该达到的安全可信度级别。

PP 是抽象层次较高的安全需求说明书，可以由产品的用户或开发者或其他第三方来定义，它为用户陈述特定的安全需要提供了一种方法。在 PP 的定义中，通常都使用 CC 中定义好的需求组件或由这些组件构成的组件包，同时，也可以使用自行定义的需求组件。在安全产品的开发过程中，PP 通常在 ST 的定义中被引用。

PP 的结构由以下几个部分组成：PP 简述、产品说明、安全环境、安全目标、安全需求、

PP 应用注释和理论依据等。

PP 简述部分给出 PP 的标识和概貌信息。产品说明部分描述将要实现 PP 所定义的安全需求的安全产品的类型和一般特性。安全环境部分描述安全产品使用环境中的有关安全因素,包括产品可能面临的安全威胁和产品的使用机构要实施的安全策略等。安全目标部分定义为解决安全环境中的各种安全问题所应确立的安全目标。安全需求部分定义安全产品为达到已确立的安全目标而应该满足的安全需求,包括安全功能需求和安全保证需求。PP 应用注释部分可有可无,它可以包含安全产品的研制、评价和使用等方面的附加支持信息。理论依据部分为以下论点提供证明依据:(1)该 PP 是一个完全的、一致的需求集合。(2)符合该 PP 要求的安全产品能在其安全环境中提供有效的安全对策。这个部分包含两个方面的内容:安全目标理论依据和安全需求理论依据。安全目标理论依据需要证明:PP 中的安全目标是从安全环境中导出的并能涵盖其中安全问题的各个方面;安全需求理论依据需要证明:PP 中的安全需求是从安全目标中导出的并能满足安全目标各个方面的要求。

3. 安全对象定义书(ST)

安全对象定义书(ST)是一份安全需求与概要设计说明书,CC 标准对它的格式有明确的规定。ST 的安全需求定义与 PP 非常相似,不同的是 ST 的安全需求是为某一特定的安全产品而定义的。ST 的安全需求可通过引用某个(或多个)PP 来定义,也可采用与定义 PP 相同的方法从头定义。ST 除包含 PP 所具有的内容外,还包含产品的概要说明。ST 的结构由以下几个部分组成:ST 简述、产品说明、安全环境、安全目标、安全需求、产品概要说明、PP 引用声明和理论依据等。

其中 ST 简述、产品说明、安全环境、安全目标和安全需求等部分与 PP 中的相应部分相似。概要说明部分对安全需求给出例化定义,一方面,针对安全功能需求定义满足这些需求的安全功能,另一方面,针对安全保证需求定义满足这些需求的安全保证措施。安全功能以非形式化的方式定义,描述要达到一定的详细程度,能把有关的实现情况表达清楚;安全保证措施可适当结合有关质量计划、生命周期计划和管理计划等加以定义。

如果 ST 中有对 PP 的引用,则 PP 引用声明部分陈述有关引用 PP 的情况,包括 ST 与 PP 间需求的一致性、ST 中对 PP 需求的进一步限定、ST 中在 PP 基础上的需求扩展等。

理论依据部分为以下论点提供证明依据:(1)该 ST 是一个完全的、一致的需求集合;(2)符合该 ST 要求的安全产品能在其安全环境中提供有效的安全对策;(3)产品概要说明涵盖了所定义的所有安全需求;(4)PP 一致性声明是有效的。这个部分包含 4 个方面的内容:安全目标理论依据、安全需求理论依据、概要说明理论依据和 PP 声明理论依据。安全目标理论依据和安全需求理论依据与 PP 中的类似。概要说明理论依据需要证明:用 ST 中设计的安全功能和安全措施去实现安全需求中的要求是合适的。如果 ST 引用了 PP,并且,ST 中的需求与 PP 不完全相同,则 PP 声明理论依据需要对其间的差别给予解释。CC 标准中安全需求的组织结构如图 8.2 所示。

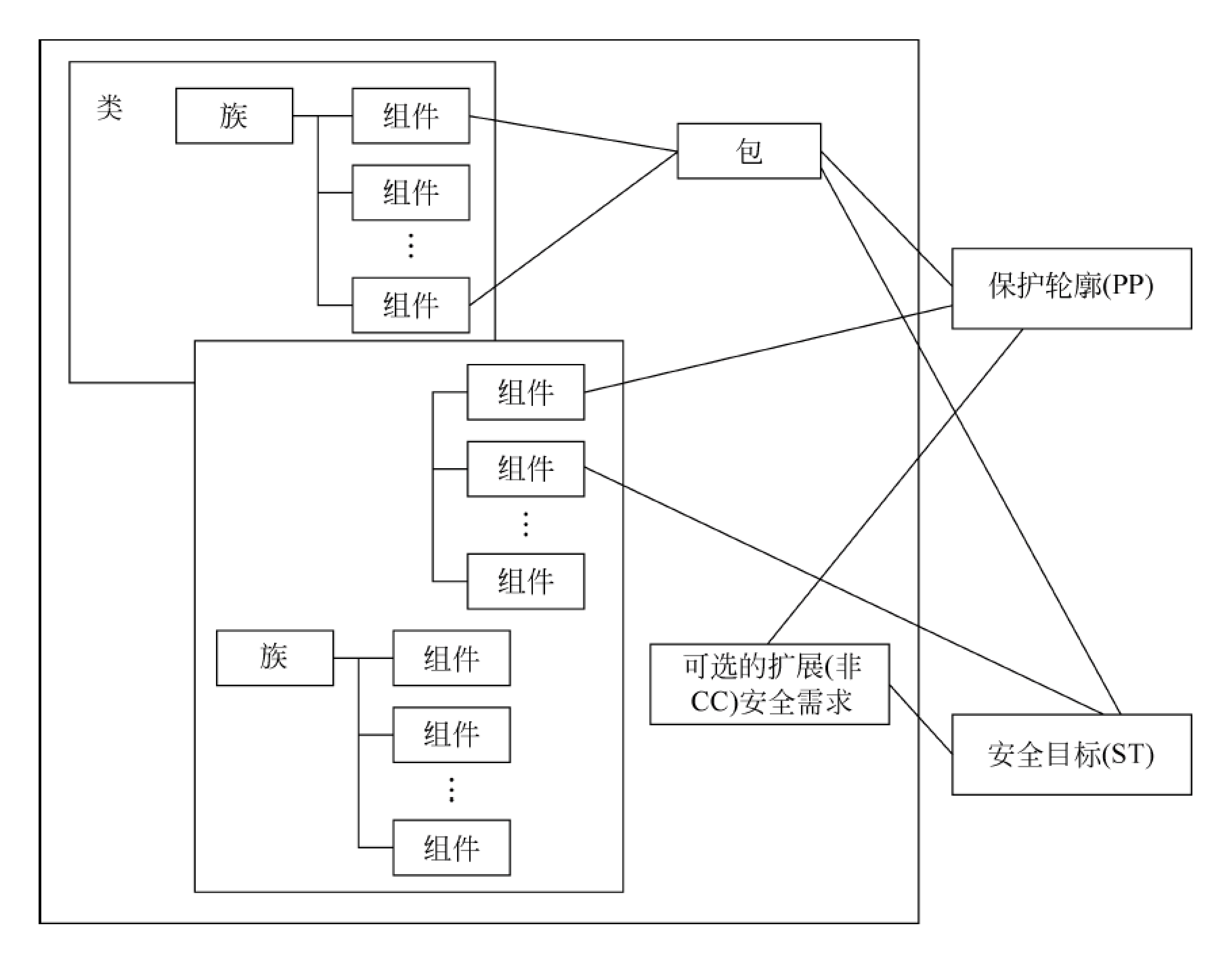


图 8.2　安全需求的组织结构

8.3.3　评估保证级别(EAL)

CC 标准定义了一套评估保证级别(Evaluation Assurance Level,EAL),作为刻画产品的安全可信度的尺度。EAL 是由 CC 中定义的安全保证需求组件构成的一个特定的组件包,由此可见,CC 对产品安全可信度的衡量是与产品的安全功能相对独立的。EAL 在产品的安全可信度与获取相应可信度的可行性及所需付出的代价之间给出了不同等级的权衡。

EAL 通过构造管理、发行与使用、开发、指南文档、生命周期支持、测试和脆弱性评估等方面所采取的措施来确立产品的安全可信度。按安全可信度由低到高依次递增的顺序,CC 定义了 7 个安全可信度级别,分别记为 EAL1、EAL2、EAL3、EAL4、EAL5、EAL6、EAL7。

EAL1 是功能测试级,它表示信息保护问题得到了适当的处理。EAL2 是结构式测试级,它要求评价时在设计信息和测试结果的提供方面得到开发人员的配合,该级提供低中级的独立安全保证。EAL3 是基于方法学的测试与检查级,它要求在设计阶段实施积极的安全工程思想,提供中级的独立安全保证。EAL4 是基于方法学的设计、测试与审查级,它要求按照良好的商业化开发惯例实施积极的安全工程思想,提供中高级的独立安全保证。EAL5 是半形式化的设计与测试级,它要求按照严格的商业化开发惯例、应用专业安全工程技术实施安全工程思想,提供高等级的独立安全保证。EAL6 是半形式化验证的设计与测试级,它通过在严格的开发环境中应用安全工程技术来获取高的安全保证,使产品能在高度危险的环境中使用。EAL7 是形式化验证的设计与测试级,它的目标是使产品能在极端危险的环境中使用,目前,该级别的实际应用只限于其安全功能可以进行广泛的形式化分析的安全产品。

CC 标准体现了软件工程与安全工程相结合的思想。信息安全产品必须按照软件工程

和安全工程的方法进行开发才能较好地获得预期的安全可信度。从需求分析到产品实现的进展角度,安全产品的开发过程可依次分为以下阶段:现实应用环境分析、确立产品安全环境、确立产品安全目标、确立产品安全需求、安全产品概要设计、安全产品实现等。一般而言,各个阶段依次顺序进行,前一个阶段的工作结果是后一个阶段的工作基础。必要时,也需要根据后面阶段工作的反馈,进一步开展前面阶段的工作,形成循环往复的过程。开发出来的产品经过安全性评价和可用性鉴定后,再投入实际使用。

从安全职能的表现形式的角度,安全产品的开发过程可依次分为以下阶段:需求组件定义、组件包定义、PP 定义、ST 定义、产品实现等。可以认为组件用于构造组件包,组件包用于构造 PP,PP 用于构造 ST,ST 用于作为产品的实现依据。但也不绝对,例如,PP 和 ST 都可以直接由组件来构造,而 PP 和 ST 又都必须引用 EAL 组件包。CC 建议尽量使用其中预定义的组件,也允许自行定义组件;CC 还允许在引用 EAL 组件包前,向该包增加其他组件,或者,把该包中的某组件替换成相应的强度更高的组件。

在 CC 中,TOE 的评价是以 ST 为基础的。不管我们通过什么方式来定义 ST,经过分解,它本质上就是通过需求组件来构造的。因此,需求组件和 PP、ST 等其他结构一起,构成了 CC 标准对信息安全产品评价的基本框架。CC 标准的评价框架面向所有信息安全产品,提供安全性评价的基本尺度和指导思想。它不限定哪类产品应该提供哪些安全功能,也不限定哪些安全功能应该具有哪个级别的安全可信度。所有这些,由产品的用户、开发人员或其他第三方在实际应用中根据实际需要来确定。

8.3.4 利用 CC 标准评估产品的一般过程

在使用 CC 标准对信息安全产品进行测试评估时,并不是直接使用 CC 来对所有类型的产品进行评估,而是引入了保护轮廓、安全目标以及功能和保证的概念。整个评估过程从概念上来说分为以下几个步骤。

首先,根据产品的类型和用户的安全需求,使用 CC 编制适用于某一类型产品(如防火墙、入侵检测系统和操作系统等产品类型)安全需求的保护轮廓,也就是保护轮廓是用户对某一类产品安全需求的标准化、结构化和规范化的描述文件。

然后,产品厂商根据保护轮廓,针对具体的安全产品编制相对应的安全目标以描述具体安全产品对用户的需求,也就是保护轮廓满足情况的描述。

当经过认可的 CC 测试实验室拿到标准化用户需求的保护轮廓、厂商所编制的具体产品的安全目标和 IT 产品后,测试实验室使用通用准则的标准评估方法检查厂商所编制的安全目标相对于安全轮廓的符合性、安全产品相对于安全目标的符合性以及安全产品的保证级别,得出正式的测试报告。在测试报告中,根据客观的测试数据,描述安全产品功能是否符合保护轮廓的要求以及安全产品的保证级别。

最后,通用准则的认可机构对测试机构的测试报告进行确认,并给安全产品厂商发放相应的认证报告。在认证报告中描述了所测试评估的安全产品功能上符合的产品类型以及安全产品的保证级别。

8.3.5 CC 的特点

CC 比起早期的评估准则,其特点体现在结构的开放性、表达方式的通用性以及结构和

表达方式的内在完备性和实用性 4 个方面。

在结构的开放性方面,CC 提出的安全功能要求和安全保证要求都可以在具体的保护轮廓和安全目标中进一步细化和扩展,例如可以增加“备份和恢复”方面的功能要求或环境安全要求。这种开放式的结构更适应信息技术和信息安全技术的发展。

通用性的特点,即给出通用的表达方式。如果用户、开发者、评估者和认可者等目标用户都使用 CC 语言,互相之间就更容易理解沟通。如果用户使用 CC 语言表达自己的安全需求,开发者就可以有针对性地描述产品和系统的安全性,评估者也更容易有效客观地进行评估,并确保评估结果对用户而言更容易理解。这种特点对规范实用方案的编写和安全性测试评估都具有重要意义。这种特点也是在经济全球化发展、全球信息化发展趋势下,进行合格评定和评估结果国家互认的需要。

CC 的这种结构和表达方式具有内在完备性和实用性的特点,具体体现在保护轮廓和安全目标的编制上。保护轮廓主要用于表达一类产品或系统的用户需求,在标准化体系中可以作为安全技术类标准对待。其内容主要包括对该类产品或系统的界定性描述,即确定要保护的对象;确定安全环境,即指明安全问题(需要保护的资产、已知的威胁、用户的组织安全策略);产品或系统的安全目的,即安全问题的相应对策(技术性和非技术性措施);信息技术安全要求,包括功能要求、保证要求和环境要求,这些要求通过满足安全目的,进一步提出具体在技术上如何解决安全问题;基本原理,指明安全要求对安全目的、安全目的对安全环境是充分必要的,以及附加的补充说明信息。

保护轮廓编制,一方面解决了技术与实际需求之间的内在完备性,另一方面用户通过分析所需要的产品和系统面临的安全问题,明确所需的安全策略,进而确定应采取的安全措施,包括技术和管理上的措施,这样有助于提高安全保护的针对性和有效性。

安全目标在保护轮廓的基础上,通过将安全要求进一步有针对性地具体化,解决了要求的具体实现。通过保护轮廓和安全目标这两种结构,便于将 CC 的安全性要求具体应用到 IT 产品的开发、生产、测试、评估和信息系统的集成、运行、评估和管理中去。

8.4　我国的信息系统安全评估标准

为了提高我国计算机信息系统的安全保障能力和防护水平,确保国家安全、公共利益和社会稳定,保障信息化建设的健康发展,1994 年 2 月,国务院发布了《中华人民共和国计算机信息系统安全保护条例》规定,要求“重点保护国家事务、国家经济建设、国防建设、国内尖端科学技术等重要领域的信息系统的安全”,同时规定计算机信息系统“实行安全等级保护”。1999 年,公安部提出并组织制定了强制性国家标准 GB 17859:1999《计算机信息安全保护等级划分准则》,该准则于 9 月 13 日经国家质量技术监督局发布,并于 2001 年 1 月 1 日起实施。该标准是建立安全等级保护制度、实施安全等级管理的重要基础性标准。它将计算机信息系统安全保护等级划分为 5 个级别,通过规范、科学和公正的评定和监督管理,一是为计算机信息系统安全等级保护管理法规的制定和执法部门的监督检查提供依据;二是为计算机信息系统安全产品的研制提供技术支持;三是为安全系统的建设和管理提供技术指导。2001 年 3 月,国家质量技术监督局发布了推荐性标准《信息技术、安全技术、信

息技术安全性评估准则》(GB/T 18336-2001),该标准等同于国际标准 ISO/IES 15408,即 CC 标准。之后,公安部于 2002 年 7 月 18 日还公布并实施了一系列计算机信息系统安全等级标准,包括 GA/T 390-2002《计算机信息系统安全等级保护通用技术要求》、GA/T 388-2002《计算机信息系统安全等级保护操作系统技术要求》、GA/T 389-2002《计算机信息系统安全等级保护数据库管理系统技术要求》等,进一步完善了计算机信息系统安全等级保护的标准体系。

8.4.1　所涉及的术语

(1) 计算机信息系统(Computer Information System):计算机信息系统是由计算机及其相关的和配套的设备、设施(含网络)构成的,按照一定的应用目标和规则对信息进行采集、加工、存储、传输、检索等处理的人机系统。

(2) 计算机信息系统可信计算基(Trusted Computing Base of Computer Information System):计算机系统内保护装置的总体,包括硬件、固件、软件和负责执行安全策略的组合体。它建立了一个基本的保护环境并提供一个可信计算系统所要求的附加用户服务。

(3) 客体(Object):信息的载体。

(4) 主体(Subject):引起信息在客体之间流动的人、进程或设备等。

(5) 敏感标记(Sensitivity Label):表示客体安全级别并描述客体数据敏感性的一组信息,可信计算基中把敏感标记作为强制访问控制决策的依据。

(6) 安全策略(Security Policy):有关管理、保护和发布敏感信息的法律、规定和实施细则。

(7) 信道(Channel):系统内的信息传输路径。

(8) 隐蔽信道(Convert Channel):允许进程以危害系统安全策略的方式传输信息的通信信道。

(9) 访问监控器(Reference Monitor):监控主体和客体之间授权访问关系的部件。

(10) 可信信道(Trusted Channel):为了执行关键的安全操作,在主体、客体及可信 IT 产品之间建立和维护的保护通信数据免遭修改和泄露的通信路径。

(11) 客体重用:在计算机信息系统可信计算基的空闲存储客体空间中,对客体初始制定、分配或再分配一个主体之前,撤销该客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时,当前主体不能获得原主体活动所产生的任何信息。

8.4.2　等级的划分及各等级的要求

《计算机信息系统安全保护等级划分准则》将信息系统划分为 5 个等级,分别是自主保护级、系统审计保护级、安全标记保护级、结构化保护级和访问验证保护级。主要的安全考核指标包括自主访问控制、强制访问控制、安全标记、身份鉴别、客体重用、审计、数据完整性、隐蔽信道分析、可信路径和可信恢复等,这些指标涵盖了不同级别的安全要求。

1. 第一级　用户自主保护级

本级的计算机信息系统可信计算基通过隔离用户与数据,使用户具备自主安全保护的能力。它具有多种形式的控制能力,对用户实施访问控制,即为用户提供可行的手段,保护用户和用户组信息,避免其他用户对数据的非法读写和破坏。本级有以下考核指标要求。

1）自主访问控制

计算机信息系统可信计算基定义和控制系统中命名用户对命名客体的访问。实施机制(例如访问控制表)允许命名用户以用户和(或)用户组的身份规定并控制客体的共享，阻止非授权用户读取敏感信息。

2）身份鉴别

计算机信息系统可信计算基初始执行时，首先要求用户标识自己的身份，并使用保护机制(口令)来鉴别用户的身份，阻止非授权用户访问用户身份鉴别数据。

3）数据完整性

计算机信息系统可信计算基通过自主完整性策略，阻止非授权用户修改或破坏敏感信息。

2. 第二级　系统审记保护级

与用户自主保护级相比，本级的计算机信息系统可信计算基实施了粒度更细的自主访问控制，它通过登录规程、审计安全相关事件和隔离资源，使用户对自己的行为负责。本级有以下考核指标的要求。

1）自主访问控制

计算机信息系统可信计算基定义和控制系统中命名用户对命名客体的访问。实施机制(例如访问控制表)允许命名用户以用户和(或)用户组的身份规定并控制客体的共享，阻止非授权用户读取敏感信息，并控制访问权限扩散。自主访问控制机制根据用户指定方式或默认方式，阻止非授权用户访问客体，访问控制的粒度是单个用户，没有存取权的用户只允许由授权用户指定对客体的访问权。

2）身份鉴别

计算机信息系统可信计算基初始执行时，首先要求用户标识自己的身份，并使用保护机制(例如口令)来鉴别用户的身份，阻止非授权用户访问用户身份鉴别数据。通过为用户提供唯一标识，计算机信息系统可信计算基能够使用户对自己的行为负责。计算机信息系统可信计算基还具备将身份标识与该用户所有可审计行为相关联的能力。

3）客体重用

在计算机信息系统可信计算基的空闲存储客体空间中，对客体初始指定、分配或再分配一个主体之前，撤销该客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时，当前主体不能获得原主体活动所产生的任何信息。

4）审计

计算机信息系统可信计算基能创建和维护受保护客体的访问审计跟踪记录，并能阻止非授权用户对它的访问和破坏。

计算机信息系统可信计算基能记录下述事件：使用身份鉴别机制；将客体引入用户地址空间(例如，打开文件、程序初始化)；删除客体；由操作员、系统管理员或(和)系统安全管理员实施的动作，以及其他与系统安全有关的事件。对于每一事件，其审计记录包括事件的日期和时间、用户、事件类型、事件是否成功。对于身份鉴别事件，审计记录包含来源(例如终端标识符)；对于客体引入用户地址空间的事件及客体删除事件，审计记录包含客体名。

对不能由计算机信息系统可信计算基独立分辨的审计事件，审计机制提供审计记录接

口,可由授权主体调用。这些审计记录区别于计算机信息系统可信计算基独立分辨的审计记录。

5）数据完整性

计算机信息系统可信计算基通过自主完整性策略,阻止非授权用户修改或破坏敏感信息。

3. 第三级　安全标记保护级

本级的计算机信息系统可信计算基具有系统审计保护级的所有功能。此外,还提供有关安全策略模型、数据标记以及主体对客体强制访问控制的非形式化描述;具有准确地标记输出信息的能力,消除通过测试发现的任何错误。本级有以下考核指标要求。

1）自主访问控制

计算机信息系统可信计算基定义和控制系统中命名用户对命名客体的访问。实施机制(例如访问控制表)允许命名用户以用户和(或)用户组的身份规定并控制客体的共享;阻止非授权用户读取敏感信息,并控制访问权限扩散。自主访问控制机制根据用户指定方式或默认方式,阻止非授权用户访问客体,访问控制的粒度是单个用户,没有存取权的用户只允许由授权用户指定对客体的访问权。阻止非授权用户读取敏感信息。

2）强制访问控制

计算机信息系统可信计算基对所有主体及其所控制的客体(例如进程、文件、段、设备)实施强制访问控制。为这些主体及客体指定敏感标记,这些标记是等级分类和非等级类别的组合,它们是实施强制访问控制的依据。计算机信息系统可信计算基支持两种或两种以上成分组成的安全级。计算机信息系统可信计算基控制的所有主体对客体的访问应满足:仅当主体安全级中等级分类高于或等于客体安全级中的等级分类,且主体安全级中的非等级类别包含了客体安全级中的全部非等级类别,主体才能读客体;仅当主体安全级中等级分类低于或等于客体安全级中的等级分类,且主体安全级中的非等级类别包含了客体安全级中的全部非等级类别,主体才能写客体。计算机信息系统可信计算基使用身份和鉴别数据,鉴别用户的身份,并保证用户创建的计算机信息系统可信计算基外部主体的安全级和授权受该用户的安全级和授权的控制。

3）标记

计算机信息系统可信计算基应维护与主体及其控制的存储客体(例如,进程、文件、段、设备)相关的敏感标记,这些标记是实施强制访问控制的基础。为了输入未加安全标记的数据,计算机信息系统可信计算基向授权用户要求并接受这些数据的安全级别,且可由计算机信息系统可信计算基审计。

4）身份鉴别

计算机信息系统可信计算基初始执行时,首先要求用户标识自己的身份,而且,计算机信息系统可信计算基维护用户身份识别数据并确定用户访问权及授权数据。计算机信息系统可信计算基使用这些数据鉴别用户,并使用保护机制(例如口令)来鉴别用户的身份,阻止非授权用户访问用户身份鉴别数据。通过为用户提供唯一标识,计算机信息系统可信计算基能够使用户对自己的行为负责。计算机信息系统可信计算基还具备将身份标识与该用户所有可审计行为相关联的能力。

5）客体重用

在计算机信息系统可信计算基的空闲存储客体空间中，对客体初始指定、分配或再分配一个主体之前，撤销该客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时，当前主体不能获得原主体活动所产生的任何信息。

6）审计

计算机信息系统可信计算基能创建和维护受保护客体的访问审计跟踪记录，并能阻止非授权的用户对它进行访问和破坏。

计算机信息系统可信计算基能记录下述事件：使用身份鉴别机制；将客体引入用户地址空间（例如，打开文件、程序初始化）；删除客体；由操作员、系统管理员或（和）系统安全管理员实施的动作，以及其他与系统安全有关的事件。对于每一事件，其审计记录包括事件的日期和时间、用户、事件类型、事件是否成功。对于身份鉴别事件，审计记录包含来源（例如终端标识符）；对于客体引入用户地址空间的事件及客体删除事件，审计记录包含客体名及客体的安全级别。此外，计算机信息系统可信计算基具有审计更改可读输出记号的能力。对不能由计算机信息系统可信计算基独立分辨的审计事件，审计机制提供审计记录接口，可由授权主体调用。这些审计记录区别于计算机信息系统可信计算基独立分辨的审计记录。

7）数据完整性

计算机信息系统可信计算基通过自主和强制完整性策略，阻止非授权用户修改或破坏敏感信息。在网络环境中，使用完整性敏感标记来确信信息在传送中未受损。

4. 第四级　结构化保护级

本级的计算机信息系统可信计算基建立于一个明确定义的形式化安全策略模型之上，它要求将第三级系统中的自主和强制访问控制扩展到所有主体和客体。此外，还要考虑隐蔽通道。本级的计算机信息系统可信计算基必须结构化为关键保护元素和非关键保护元素。计算机信息系统可信计算基的接口也必须明确定义，使其设计与实现能经受更充分的测试和更完整的复审。加强了鉴别机制，支持系统管理员和操作员的职能；提供可信设施管理，增强了配置管理控制。系统具有相当的抗渗透能力。本级有以下考核指标要求。

1）自主访问控制

计算机信息系统可信计算基定义和控制系统中命名用户对命名客体的访问，实施机制（例如访问控制表）允许命名用户以用户和（或）用户组的身份规定并控制客体的共享；阻止非授权用户读取敏感信息，并控制访问权限扩散。自主访问控制机制根据用户指定方式或默认方式，阻止非授权用户访问客体，访问控制的粒度是单个用户，没有存取权的用户只允许由授权用户指定对客体的访问权。

2）强制访问控制

计算机信息系统可信计算基对所有主体及其所控制的客体（例如进程、文件、段、设备）实施强制访问控制。为这些主体及客体指定敏感标记，这些标记是等级分类和非等级类别的组合，它们是实施强制访问控制的依据。计算机信息系统可信计算基支持两种或两种以上成分组成的安全级。计算机信息系统可信计算基外部的所有主体对客体的直接或间接的访问应满足：仅当主体安全级中等级分类高于或等于客体安全级中的等级分类，且主体安全级中的非等级类别包含了客体安全级中的全部非等级类别，主体才能读客体；仅当主体安全级中等级分类低于或等于客体安全级中的等级分类，且主体安全级中的非等级类别包

含了客体安全级中的全部非等级类别，主体才能写客体。计算机信息系统可信计算基使用身份和鉴别数据鉴别用户的身份，并保证用户创建的计算机信息系统可信计算基外部主体的安全级和授权受该用户的安全级和授权的控制。

3）标记

计算机信息系统可信计算基应维护与外部主体直接或间接访问到的计算机信息系统资源（例如，主体、存储客体、只读存储器）相关的敏感标记。这些标记是实施强制访问控制的基础。为了输入未加安全标记的数据，计算机信息系统可信计算基向授权用户要求并接受这些数据的安全级别，且可由计算机信息系统可信计算基审计。

4）身份鉴别

计算机信息系统可信计算基初始执行时，首先要求用户标识自己的身份，而且，计算机信息系统可信计算基维护用户身份识别数据并确定用户访问权及授权数据。计算机信息系统可信计算基使用这些数据鉴别用户身份，并使用保护机制（例如口令）来鉴别用户的身份，阻止非授权用户访问用户身份鉴别数据。通过为用户提供唯一标识，计算机信息系统可信计算基能够使用户对自己的行为负责。计算机信息系统可信计算基还具备将身份标识与该用户所有可审计行为相关联的能力。

5）客体重用

在计算机信息系统可信计算基的空闲存储客体空间中，对客体初始指定、分配或再分配一个主体之前，撤销该客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时，当前主体不能获得原主体活动所产生的任何信息。

6）审计

计算机信息系统可信计算基能创建和维护受保护客体的访问审计跟踪记录，并能阻止非授权的用户对它访问和破坏。

计算机信息系统可信计算基能记录下述事件：使用身份鉴别机制；将客体引入用户地址空间（例如，打开文件、程序初始化）；删除客体；由操作员、系统管理员或（和）系统安全管理员实施的动作，以及其他与系统安全有关的事件。对于每一事件，其审计记录包括事件的日期和时间、用户、事件类型、事件是否成功。对于身份鉴别事件，审计记录包含请求的来源（例如终端标识符）；对于客体引入用户地址空间的事件及客体删除事件，审计记录包含客体名及客体的安全级别。此外，计算机信息系统可信计算基具有审计更改可读输出记号的能力。

对不能由计算机信息系统可信计算基独立分辨的审计事件，审计机制提供审计记录接口，可由授权主体调用。这些审计记录区别于计算机信息系统可信计算基独立分辨的审计记录。

计算机信息系统可信计算基能够审计利用隐蔽存储信道时可能被使用的事件。

7）数据完整性

计算机信息系统可信计算基通过自主和强制完整性策略，阻止非授权用户修改或破坏敏感信息。在网络环境中，使用完整性敏感标记来确信信息在传送中未受损。

8）隐蔽信道分析

系统开发者应彻底搜索隐蔽存储信道，并根据实际测量或工程估算确定每一个被标识信道的最大带宽。

9）可信路径

对用户的初始登录和鉴别，计算机信息系统可信计算基在它和用户之间提供可信通信路径。该路径上的通信只能由该用户初始化。

5. 访问验证保护级

本级的计算机信息系统可信计算基满足访问监控器需求。访问监控器仲裁主体对客体的全部访问。访问监控器本身是抗篡改的；必须足够小，能够分析和测试。为了满足访问监控器需求，计算机信息系统可信计算基在其构造时，排除那些对实施安全策略来说非必要的代码；在设计和实现时，从系统工程角度将其复杂性降低到最小程度。支持安全管理员职能；扩充审计机制，当发生与安全相关的事件时发出信号；提供系统恢复机制。系统具有很高的抗渗透能力。本级有以下考核指标的要求。

1）自主访问控制

计算机信息系统可信计算基定义和控制系统中命名用户对命名客体的访问。实施机制（例如访问控制表）允许命名用户以用户和（或）用户组的身份规定并控制客体的共享；阻止非授权用户读取敏感信息，并控制访问权限扩散。

自主访问控制机制根据用户指定方式或默认方式，阻止非授权用户访问客体，访问控制的粒度是单个用户，访问控制能够为每个命名客体指定命名用户和用户组，并规定它们对客体的访问模式。没有存取权的用户只允许由授权用户指定对客体的访问权。

2）强制访问控制

计算机信息系统可信计算基对外部主体能够直接或间接访问的所有资源（例如主体、存储客体和输入输出资源）实施强制访问控制。为这些主体及客体指定敏感标记，这些标记是等级分类和非等级类别的组合，它们是实施强制访问控制的依据。计算机信息系统可信计算基支持两种或两种以上成分组成的安全级。计算机信息系统可信计算基外部的所有主体对客体的直接或间接的访问应满足：仅当主体安全级中等级分类高于或等于客体安全级中的等级分类，且主体安全级中的非等级类别包含了客体安全级中的全部非等级类别，主体才能读客体；仅当主体安全级中等级分类低于或等于客体安全级中的等级分类，且主体安全级中的非等级类别包含了客体安全级中的全部非等级类别，主体才能写客体。计算机信息系统可信计算基使用身份和鉴别数据，鉴别用户的身份，并保证用户创建的计算机信息系统可信计算基外部主体的安全级和授权受该用户的安全级和授权的控制。

3）标记

计算机信息系统可信计算基应维护与可被外部主体直接或间接访问到的计算机信息系统资源（例如，主体、存储客体、只读存储器）相关的敏感标记。这些标记是实施强制访问控制的基础。为了输入未加安全标记的数据，计算机信息系统可信计算基向授权用户要求并接受这些数据的安全级别，且可由计算机信息系统可信计算基审计。

4）身份鉴别

计算机信息系统可信计算基初始执行时，首先要求用户标识自己的身份，而且，计算机信息系统可信计算基维护用户身份识别数据并确定用户访问权及授权数据。计算机信息系统可信计算基使用这些数据鉴别用户，并使用保护机制（例如口令）来鉴别用户的身份，阻止非授权用户访问用户身份鉴别数据。通过为用户提供唯一标识，计算机信息系统可信计算基能够使用户对自己的行为负责。计算机信息系统可信计算基还具备将身份标识与该用户

所有可审计行为相关联的能力。

5）客体重用

在计算机信息系统可信计算基的空闲存储客体空间中，对客体初始指定、分配或再分配一个主体之前，撤销该客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时，当前主体不能获得原主体活动所产生的任何信息。

6）审计

计算机信息系统可信计算基能创建和维护受保护客体的访问审计跟踪记录，并能阻止非授权的用户对它进行访问和破坏。

计算机信息系统可信计算基能记录下述事件：使用身份鉴别机制；将客体引入用户地址空间（例如，打开文件、程序初始化）；删除客体；由操作员、系统管理员或（和）系统安全管理员实施的动作，以及其他与系统安全有关的事件。对于每一事件，其审计记录包括事件的日期和时间、用户、事件类型、事件是否成功。对于身份鉴别事件，审计记录包含来源（例如终端标识符）；对于客体引入用户地址空间的事件及客体删除事件，审计记录包含客体名及客体的安全级别。此外，计算机信息系统可信计算基具有审计更改可读输出记号的能力。

对不能由计算机信息系统可信计算基独立分辨的审计事件，审计机制提供审计记录接口，可由授权主体调用。这些审计记录区别于计算机信息系统可信计算基独立分辨的审计记录。计算机信息系统可信计算基能够审计利用隐蔽存储信道时可能被使用的事件。

计算机信息系统可信计算基包含能够监控可审计安全事件发生与积累的机制，当超过阈值时，能够立即向安全管理员发出警报。并且，如果这些与安全相关的事件继续发生或积累，系统应以最小的代价终止它们。

7）数据完整性

计算机信息系统可信计算基通过自主完整性策略，阻止非授权用户修改或破坏敏感信息。在网络环境中，使用完整性敏感标记来确信信息在传送中未受损。

8）隐蔽信道分析

系统开发者应彻底搜索隐蔽信道，并根据实际测量或工程估算确定每一个被标识信道的最大带宽。

9）可信路径

当连接用户时（例如注册、更改主体安全级），计算机信息系统可信计算基提供它与用户之间的可信通信路径。可信路径上的通信只能由该用户或计算机信息系统可信计算基激活，在逻辑上与其他路径上的通信相隔离，且能正确地加以区分。

10）可信恢复

计算机信息系统可信计算基提供过程和机制，保证计算机信息系统能够在失效或中断后，进行不损害任何安全保护性能的恢复。

8.5 信息安全等级保护

当前，我国已建成规模宏大、覆盖全国的信息网络，国家重大工程建设的重要网络和信息系统，已成为支撑国民经济和社会发展的关键基础设施。为了保障我国重要网络和系统

的安全，党中央国务院作出了一系列重大决策和部署，确立了以等级保护制度为核心的信息安全保障体系。

1994 年国务院颁发的《中华人民共和国计算机信息系统安全保护条例》规定：“计算机信息系统实行等级保护，安全等级的划分标准和安全等级保护的具体方法，由公安部会同有关部门制定”。为此，以强制性国家标准 GB 17859-1999《计算机信息系统安全保护等级划分准则》(以下简称准则)为核心的一系列等级保护国标，于 1999 年经国家质量技术监督局批准发布，于 2001 年 1 月起实施。有关《准则》的具体内容 8.4 节已经做了详细介绍。

2004 年公安部发布《关于信息安全等级保护工作的实施意见》，明确指出信息安全等级保护制度的基本内容是：对国家秘密信息、法人和其他组织及公民的专有信息以及公开信息和存储、传输、处理这些信息的信息系统分等级实行安全保护，对信息系统中使用的信息安全产品实行按等级管理，对信息系统中发生的信息安全事件分等级响应、处置。

此外，2007 年公安部发布了《信息安全等级保护管理办法》，明确了信息安全等级保护的基本内容、流程及工作要求，明确了信息系统运营使用单位和主管部门、监管部门在信息安全等级保护工作中的职责、任务，为开展信息安全等级保护工作提供了规范保障。

对信息系统的安全等级划分有两种描述形式，《准则》根据安全保护能力将信息系统划分为 5 个安全等级，即用户自主保护级、系统审计保护级、安全标记保护级、结构化保护级、访问验证保护级；《管理办法》根据主体遭受破坏后对客体的破坏程度划分为 5 个安全等级，即自主保护级、指导保护级、监督保护级、强制保护级、专控保护级。实际上，这两种安全等级具有对应关系。

8.5.1　等级保护的基本概念

信息系统安全等级保护是指对信息安全实行等级化保护和等级化管理。根据信息系统应用业务重要程度及其实际安全需求，实行分级、分类、分阶段保护，保障信息安全和系统安全正常运行，维护国际利益、公共利益和社会稳定。

8.5.2　等级保护的定级要素及级别划分

信息系统的安全保护等级应当根据信息系统在国家安全、经济建设、社会生活中的重要程度，遭到破坏后对国家安全、社会秩序、公共利益以及公民、法人和其他组织的合法权益的危害程度等因素确定。

信息系统的安全保护等级由两个定级要素决定：等级保护对象受到破坏时所侵害的客体和对客体造成侵害的程度。

1. 受侵害的客体

等级保护对象受到破坏时所侵害的客体包括以下三个方面：

(1) 公民、法人和其他组织的合法权益；

(2) 社会秩序、公共利益；

(3) 国家安全。

2. 对客体的侵害程度

(1) 一般损害；

(2) 严重损害;

(3) 特别严重损害。

根据以上要素,《管理办法》将信息系统等级保护分为5级,如表8.2所示,从第1级到第5级逐级增高,确定为第3级以上的信息系统属于国家重要信息系统。

(1) 第1级:自主保护级,适用于小型私营、个体企业、中小学、乡镇所属信息系统、县级单位中一般的信息系统。

(2) 第2级:指导保护级,适用于县级某些单位中的重要信息系统、地市级以上国家机关、企事业单位内部一般的信息系统,例如非涉及工作秘密、商业秘密、敏感信息的办公系统和管理系统等。

(3) 第3级:监督保护级,一般适用于地市级以上国家机关、企事业单位内部重要的信息系统。跨省或全国联网运行的用于生产、调度、管理、指挥、作业、控制等方面的重要信息系统,以及这类系统在省、地市的分支系统,中央各部委、省门户重要网站、跨省连接的网络系统等,例如网上银行系统、证券集中交易系统、海关通关系统、民航离港控制系统等为三级信息系统。

(4) 第4级:强制保护级。一般适用于国家重要领域、部门中涉及国计民生、国家利益、国家安全、影响社会稳定的核心系统,例如电信骨干传输网、电力能量管理系统、银行核心业务系统、铁路客票系统、列车指挥调度系统等。

(5) 第5级:专控保护级,适用于国家特殊领域的极端重要系统。

表8.2 信息系统等级保护划分

等级	对象	侵害客体	侵害程度	监督强度
第1级	一般系统	合法权益	损害	自主保护
第2级		合法权益	严重损害	指导
		社会秩序和公共利益	损害	
第3级	重要系统	社会秩序和公共利益	严重损害	监督检查
		国家安全	损害	
第4级		社会秩序和公共利益	特别严重损害	强制监督检查
		国家安全	严重损害	
第5级	极端重要系统	国家安全	特别严重损害	专门监督检查

8.5.3 等级保护工作的环节

等级保护工作的环节主要包括信息系统定级、备案、安全建设整改、等级测评、监督检查。

(1) 定级:对信息系统进行定级是等级保护的基础,具体是指各单位、各部门按照等级保护有关政策和标准要求,确定信息系统的安全保护等级,组织专家进行评审,主管部门审批。

(2) 备案:是指信息系统等级确定后,第2级以上的信息系统到公安机关备案,公安机关审核信息系统等级和有关材料,符合要求的办理备案证明。

(3) 安全建设整改:备案单位根据信息系统安全保护等级,按照等级保护有关政策和标准要求,开展安全建设整改,建设安全设施,落实安全措施,落实安全责任,建立并落实安

全管理制度。

(4) 等级测评：备案单位选择《全国信息安全等级保护测评机构推荐目录》中的测评机构开展等级测评，对照等级保护有关标准，查找安全问题和差距，为开展安全建设整改提供依据。

(5) 监督检查：备案单位定期开展自查，行业主管部门组织开展督导检查，公安机关依法对各单位、各部门开展等级保护工作情况进行定期检查。

8.6 小　　结

为了对现有计算机系统进行统一评价，为计算机系统制造商提供一个权威的系统安全性标准，需要有一个计算机系统安全评测标准。本章主要介绍了具有广泛影响力的美国国防部推出的 TCSEC 标准、国际标准化组织 ISO/IEC JTC1 发布的 CC 标准，以及我国的信息系统安全评估标准。同时，为了保障我国重要网络和系统的安全，需要对信息安全实行等级化保护和等级化管理。

习　　题

一、填空题

1. 《可信计算机系统评估准则(TCSEC)》将系统的安全等级分为________________，________________，________________和________________ 4 类。

2. 目前通用 Windows 操作系统满足 TCSEC 的________安全性要求。

3. CC 分为三个部分，分别是________、________、________。

4. 《计算机信息系统安全保护等级划分准则》将信息系统划分为 5 个等级，分别为________、________、________、________和________。

5. 在 TCSEC 中对隐蔽信道首次提出要求的安全级别是________。

6. 在 TCSEC 中对首次引入审计的安全级别是________。

7. 我国信息安全等级保护划分为 5 级，分别是________、________、________、________、________。

8. 信息系统的等级保护工作分为 5 个环节，分别是________、________、________、________、________。

二、简答题

1. 本章中介绍的一些安全标准有何联系与区别？

2. 我国对信息系统的安全等级划分通常有两种描述形式，即根据安全保护能力划分安全等级的描述，以及根据主体遭受破坏后对客体的破坏程度划分安全等级的描述。谈谈这两种等级划分的对应关系。

3. 知识拓展：查阅资料，详细了解国家信息安全等级保护政策和标准内容。

第 9 章　信息系统安全风险评估

信息系统安全问题单凭技术是无法得到彻底解决的，它的解决涉及到政策法规、管理、标准、技术等方方面面，任何单一层次上的安全措施都不可能提供真正全方位的安全，信息系统安全问题的解决更应该站在系统工程的角度来考虑。在这项系统工程中，信息系统安全风险评估占有重要的地位，它是信息系统安全的基础和前提。

本章 9.1 节简单介绍风险评估的概念，9.2 节介绍定量、定性等风险评估的方法，9.3 节介绍目前常用的风险评估的工具，9.4 节详细介绍风险评估的过程，9.5 节指出风险评估存在的问题。

9.1　风险评估简介

信息系统风险评估是从风险管理的角度，运用科学的方法和手段，系统分析网络和信息系统所面临的威胁及其存在的脆弱性，评估安全事件一旦发生可能造成的危害程度，为防范和化解信息安全风险，或者将风险控制在可接受的水平，制定有针对性的抵御威胁的防护对策和整改措施，以最大限度地为保障计算机网络信息系统安全提供科学依据。

在信息化建设中，各类信息系统由于可能存在软硬件设备缺陷、系统集成缺陷等，以及信息安全管理中潜在的薄弱环节，都将导致不同程度的安全风险。

对系统进行风险分析和评估的目的是了解系统目前与未来的风险所在，评估这些风险可能带来的安全威胁与影响程度，为安全策略的确定、信息系统的建立及安全运行提供依据。同时通过第三方权威或者国际机构评估和认证，也给用户提供了信息技术产品和系统可靠性的信心，增强产品、单位的竞争力。

信息系统风险分析和评估是一个复杂的过程，一个完善的信息安全风险评估架构应该具备相应的标准体系、技术体系、组织架构、业务体系和法律法规。

我国的信息安全风险评估工作在有条不紊地推进。2003 年 7 月，中办发[2003]27 号文件对开展信息、安全风险评估工作提出了明确的要求。国信办委托国家信息中心牵头，成立了国家信息安全风险评估课题组对信息安全风险评估相关工作展开调查研究。2004 年 3 月 29 日正式启动了信息安全风险评估标准草案的编制工作。2004 年年底完成了《信息安全风险评估指南》标准草案。2005 年由国务院信息办组织在北京、上海、黑龙江、云南，由人民银行国家税务总局、国家信息中心与国家电力总公司开展了验证《信息安全风险评估指南》的可行性与可用性的试点工作。2006 年 6 月 19 日，全国信息安全标准化技术委员会经过讨论，将标准正式命名为《信息安全技术　信息安全风险评估规范》，并同意通过评审。由国家标准化管理委员会审查批准发布的 GB/T 20984-2007《信息安全技术信息安全风险评估规范》于 2007 年 11 月 1 日正式实施。

《信息安全风险评估规范》(以下简称《规范》)是我国开展信息安全风险评估工作遵循的

国家标准。《规范》定义了风险评估的基本概念、原理及实施流程，对被评估系统的资产、威胁和脆弱性识别要求进行了详细描述，并给出了具体的定级依据；提出了风险评估在信息系统生命周期不同阶段的实施要点，以及风险评估的工作形式。

9.2　风险评估的方法

在评估过程中使用何种方法对评估的有效性占有举足轻重的地位。评估方法的选择直接影响到评估过程中的每个环节，甚至可以左右最终的评估结果，所以需要根据系统的具体情况，选择合适的风险评估方法。风险评估的方法有很多种，概括起来可分为三大类：定量的风险评估方法、定性的风险评估方法、定性与定量相结合的评估方法。

9.2.1　定量评估方法

定量的评估方法是指运用数量指标来对风险进行评估，通过对风险相关的所有要素（资产价值、威胁频率、弱点利用程度、安全措施的效率和成本等）赋值实现对风险评估结果的量化。典型的定量分析方法有因子分析法、聚类分析法、时序模型、回归模型、等风险图法、决策树法等。定量评估中涉及的几个重要概念如下。

（1）暴露因子（Exposure Factor，EF）：特定威胁对特定资产造成损失的百分比，即损失的程度。

（2）单一损失期望（Single Loss Expectancy，SLE）：即特定威胁可能造成的潜在损失总量。

（3）年度发生率（Annualized Rate of Occurrence，ARO）：威胁在一年内估计会发生的频率。

（4）年度损失期望（Annualized Loss Expectancy，ALE）：即特定资产在一年内遭受损失的预期值。

定量分析的过程如下：

（1）识别资产并为资产赋值。

（2）通过威胁和弱点评估，评估特定威胁作用于特定资产所造成的影响，即确定 EF（取值为 0%～100%）。

（3）计算特定威胁发生的概率，即 ARO。

（4）计算资产的 SLE：SLE＝总资产值×EF。

（5）计算资产的 ALE：ALE＝SLE×ARO。

【例 9-1】　假定某公司投资 500 000 美元建了一个网络运营中心，其最大威胁是火灾，一旦火灾发生，网络运营中心的估计损失程度 EF＝45%，根据消防部门推断，该网络运营中心所在的地区每 5 年会发生一次火灾，于是得出 ARO＝0.2。基于以上数据，该公司网络运营中心的 ALE 将是 500 000×45%×0.2＝45 000 美元。

可以看到，对定量分析来说，EF 和 ARO 两个指标最为关键。

定量评估方法的优点是用客观、直观的数据表述评估的结果，看起来一目了然，有时，一个数据所能够说明的问题可能是用一大段文字也不能够阐述清楚的；并且定量分析方法的

采用，可以使研究结果更科学、更严密、更深刻。缺点是常常为了量化，使本来比较复杂的事物简单化、模糊化了，有的风险因素被量化以后还可能被误解和曲解。

9.2.2 定性评估方法

定性评估方法是主要依据研究者的知识、经验、历史教训、政策走向及特殊变例等非量化资料对系统风险状况做出判断的过程。它主要以与调查对象的深入访谈做出的个案记录为基本资料，然后通过一个理论推导演绎的分析框架，对资料进行编码整理，在此基础上做出调查结论。典型的定性分析方法有因素分析法、逻辑分析法、历史比较法、德尔斐法。

定性评估方法的优点是避免了定量方法的缺点，可以挖掘出一些蕴藏很深的思想，使评估的结论更全面、更深刻；但它的主观性很强，往往需要凭借分析者的经验和直觉，或者业界的标准和惯例，对评估者本身的要求很高。定性评估方法为风险管理各要素（资产价值、威胁的可能性、脆弱点被利用的容易度等）的大小或高低程度定性分级，例如高、中、低三级。

与定量分析方法相比，定性分析较为主观，精确性不够，而定量分析较为客观，比较精确。此外，定量分析的结果直观，容易理解，而定性分析的结果则很难有统一的解释。

9.2.3 定性与定量相结合的综合评估方法

系统风险评估是一个复杂的过程，需要考虑的因素很多，有些评估要素可以用量化的形式来表达，而对有些要素的量化又是很困难甚至是不可能的，所以我们不主张在风险评估过程中一味地追求量化，也不认为一切量化的风险评估过程都是科学、准确的。我们认为定量分析是定性分析的基础和前提，定性分析应建立在定量分析的基础上才能揭示客观事物的内在规律。定性分析则是灵魂，是形成概念、观点、做出判断、得出结论所必须依靠的，在复杂的信息系统风险评估过程中，不能将定性分析和定量分析两种方法简单地割裂开来，而是应该将这两种方法融合起来，采用综合的评估方法。

9.2.4 典型的风险评估方法

在信息系统风险评估过程中，层次分析法（AHP）经常被用到，它是一种综合的评估方法。该方法是由美国著名的运筹学专家 SattyTL 于 20 世纪 70 年代提出来的，是一种定性与定量相结合的多目标决策分析方法。这一方法的核心是将决策者的经验判断给予量化，从而为决策者提供定量形式的决策依据。目前该方法已被广泛地应用于尚无统一度量标尺的复杂问题的分析，解决用纯参数数学模型方法难以解决的决策分析问题。该方法对系统进行分层次、拟定量、规范化处理，在评估过程中经历系统分解、安全性判断和综合判断三个阶段。它的基本步骤如下。

(1) 系统分解，建立层次结构模型。层次模型的构造基于分解法的思想，进行对象的系统分解。它的基本层次有三类：目标层、准则层和指标层，目的是基于系统基本特征建立系统的评估指标体系。

(2) 构造判断矩阵，通过单层次计算进行安全性判断。判断矩阵的作用是在上一层某一元素约束条件下，对同层次的元素间的相对重要性进行比较，根据心理学家提出的“人区分信息等级的极限能力为 7 ± 2”的研究结论，AHP 方法在对评估指标的相对重要程度进行测量时，引入了九分位的相对重要的比例标度，构成判断矩阵。计算的中心问题是求解判断

矩阵的最大特征根及其对应的特征向量；通过判断矩阵及矩阵运算的数学方法，确定对于上一层次的某个元素而言，本层次中与其相关元素的相对风险权值。

(3) 层次总排序，完成综合判断。计算各层元素对系统目标的合成权重，完成综合判断，进行总排序，以确定递阶结构图中最底层各个元素在总目标中的风险程度。

9.3　风险评估的工具

风险评估工具是风险评估的辅助手段，是保证风险评估结果可信度的一个重要因素。风险评估工具的使用不但在一定程度上解决了手动评估的局限性，最主要的是它能够将专家知识进行集中，使专家的经验知识被广泛应用。

根据风险评估过程中的主要任务和作用原理的不同，风险评估工具可以分为风险评估与管理工具、系统基础平台风险评估工具、风险评估辅助工具三类。风险评估与管理工具是一套集成了风险评估各类知识和判断依据的管理信息系统，以规范风险评估的过程和操作方法；或者是用于收集评估所需要的数据和资料，基于专家经验，对输入输出进行模型分析。系统基础平台风险评估工具主要用于对信息系统的主要部件（如操作系统、数据库系统、网络设备等）的脆弱性进行分析，或实施基于脆弱性的攻击。风险评估辅助工具则实现对数据的采集、现状分析和趋势分析等单项功能，为风险评估各要素的赋值、定级提供依据。

1. 风险评估与管理工具

风险评估与管理工具大部分是基于某种标准方法或某组织自行开发的评估方法，可以有效地通过输入数据来分析风险，给出对风险的评价并推荐控制风险的安全措施。

风险评估与管理工具通常建立在一定的模型或算法之上，风险由重要资产、所面临的威胁以及威胁所利用的脆弱性三者来确定；也有的通过建立专家系统，利用专家经验进行分析，给出专家结论。这种评估工具需要不断进行知识库的扩充。

此类工具实现了对风险评估全过程的实施和管理，包括被评估信息系统基本信息获取、资产信息获取、脆弱性识别与管理、威胁识别、风险计算、评估过程与评估结果管理等功能。评估的方式可以通过问卷的方式，也可以通过结构化的推理过程，建立模型，输入相关信息，得出评估结论。通常这类工具在对风险进行评估后都会有针对性地提出风险控制措施。

根据实现方法的不同，风险评估与管理攻击可以分为以下三类。

1）基于信息安全标准的风险评估与管理工具

目前国际上存在多种不同的风险分析标准和指南，不同的风险方法侧重点不同。以这些标准或指南的内容为基础，分别开发相应的评估工具，完成遵循标准或指南的风险评估过程。

2）基于知识的风险评估与管理工具

基于知识的风险评估与管理工具并不仅仅遵循某个单一的标准或指南，而是将各种风险分析方法进行综合，并结合实践经验，形成风险评估知识库，以此为基础完成综合评估。它还涉及来自类似组织（包括规模、商务目标和市场等）的最佳实践，主要通过多种途径采集相关信息，识别组织的风险和当前的安全措施；与特定的标准或最佳实践进行比较，从中找

出不符合的地方，按照标准或最佳实践的推荐选择安全措施以控制风险。

3）基于模型的风险评估与管理工具

基于标准或基于知识的风险评估与管理工具，都使用了定性分析方法或定量分析方法，或者将定性与定量相结合。定性分析方法是目前广泛采用的方法，需要凭借评估者的知识、经验和直觉，或者业界的标准和实践，为风险的各个要素定级。定性分析法操作相对容易，但也可能因为评估者经验和直觉的偏差而使分析结果失准。定量分析则对构成风险的各个要素和潜在损失水平赋予数值或货币金额，通过对度量风险的所有要素进行赋值，建立综合评价的数学模型，从而完成风险评估的量化计算。定量分析方法准确，但前期建立系统风险模型较困难。定性与定量结合分析方法就是将风险要素的赋值和计算，根据需要分别采取定性和定量的方法完成。基于模型的风险评估与管理工具是在对系统各组成部分、安全要素充分研究的基础上，对典型系统的资产、威胁、脆弱性建立量化或半量化的模型，根据采集信息的输入，得到评价的结果。

2. 系统基础平台风险评估工具

系统基础平台风险评估工具包括脆弱性扫描工具和渗透性测试工具。脆弱性扫描工具又称为安全扫描器、漏洞扫描仪等，主要用于识别网络、操作系统、数据库系统的脆弱性。通常情况下，这些工具能够发现软件和硬件中已知的脆弱性，以决定系统是否易受已知攻击的影响。

脆弱性扫描工具是目前应用最广泛的风险评估工具，主要完成操作系统、数据库系统、网络协议、网络服务等的安全脆弱性检测功能，目前常见的脆弱性扫描工具有以下几种类型。

（1）基于网络的扫描器：在网络中运行，能够检测如防火墙错误配置或连接到网络上的易受攻击的网络服务器的关键漏洞。

（2）基于主机的扫描器：发现主机的操作系统、特殊服务和配置的细节，发现潜在的用户行为风险，如密码强度不够，也可实施对文件系统的检查。

（3）分布式网络扫描器：由远程扫描代理、对这些代理的即插即用更新机制、中心管理点三部分构成，用于企业级网络的脆弱性评估，分布和位于不同的位置、城市甚至不同的国家。

（4）数据库脆弱性扫描器：对数据库的授权、认证和完整性进行详细的分析，也可以识别数据库系统中潜在的脆弱性。

渗透性测试工具是根据脆弱性扫描工具扫描的结果进行模拟攻击测试，判断被非法访问者利用的可能性。这类工具通常包括黑客工具、脚本文件。渗透性测试的目的是检测已发现的脆弱性是否真的会给系统或网络带来影响。通常渗透性工具与脆弱性扫描工具一起使用，并可能会对被评估系统的运行带来一定影响。

3. 风险评估辅助工具

科学的风险评估需要大量的实践和经验数据的支持，这些数据的积累是风险评估科学性的基础。风险评估过程中，可以利用一些辅助性的工具和方法来采集数据，帮助完成现状分析和趋势判断，列举如下。

（1）检查列表：检查列表是基于特定标准或基线建立的，对特定系统进行审查的项

目条款。通过检查列表，操作者可以快速定位系统目前的安全状况与基线要求之间的差距。

（2）入侵检测系统：入侵检测系统通过部署检测引擎，收集、处理整个网络中的通信信息，以获取可能对网络或主机造成危害的入侵攻击事件；帮助检测各种攻击试探和误操作，同时也可以作为一个警报器，提醒管理员发生的安全状况。

（3）安全审计工具：用于记录网络行为，分析系统或网络安全现状，它的审计记录可以作为风险评估中的安全现状数据，并可用于判断被评估对象威胁信息的来源。

（4）拓扑发现工具：通过接入点接入被评估网络，完成被评估网络中的资产发现功能，并提供网络资产的相关信息，包括操作系统版本、型号等。拓扑发现工具的主要功能是自动完成网络硬件设备的识别、发现。

（5）资产信息收集系统：通过提供调查表形式，完成被评估信息系统数据、管理、人员等资产信息的收集功能，了解到组织的主要业务、重要资产、威胁、管理上的缺陷、采用的控制措施和安全策略的执行情况。此类系统主要采取电子调查表形式，需要被评估管理系统人员参与填写，并自动完成资产信息获取。

（6）其他：如用于评估过程参考的评估指标库、知识库、漏洞库、算法库、模型库等。

风险评估最常用的还是一些专用的自动化风险评估工具，下面介绍几款典型的风险评估工具。

9.3.1　SAFESuite 套件

SAFESuite 套件是 Internet Security Systems（简称 ISS）公司开发的网络脆弱点检测软件，它由 Internet 扫描器、系统扫描器、数据库扫描器、实时监控和 SAFESuite 套件决策软件构成，是一个完整的信息系统评估系统。

9.3.2　WebTrends Security Analyzer 套件

WebTrends Security Analyzer 套件主要针对 Web 站点安全的检测和分析软件，它是 NetIQ-WebTrends 公司的系列产品。其系列产品为企业提供一套完整的、可升级的、模块式的、易于使用的解决方案。产品系列包括 WebTrends Reporting Center、Analysis Suite、WebTrends Log Analyzer、Security Analyzer、WebTrends、Firewall Suite and WebTrends Live 等，它可以找出大量隐藏在 Linux 和 Windows 服务器、防火墙、路由器等软件中的威胁和脆弱点，并可针对 Web 和防火墙日志进行分析，由它生成的 HTML 格式的报告被认为是目前市场上做得最好的。报告里对找到的每个脆弱点进行了说明，并根据脆弱点的优先级进行了分类，还包括一些消除风险、保护系统的建议。

9.3.3　Cobra

Cobra（Consultative Objective and Bi-funchtional Risk Analysis）是一套专门用于进行风险分析的工具软件，其中也包含促进安全策略执行、外部安全标准（ISO 17799）评定的功能模块。用 Cobra 进行风险分析时，分三个步骤：调查表生成、风险调查、报告生成。Cobra 的操作过程简单而灵活，安全分析人员只需要清楚当前的信息系统状况，并对之作出正确的解释即可，所有烦琐的分析工作都交由 Cobra 来自动完成。

9.3.4 CC tools

CC tools 是针对 CC 开发的工具，它帮助用户按照 CC 标准自动生成 PP（保护轮廓）和 ST（安全目标）报告。

以上这些工具有的是通过技术手段，如漏洞扫描、入侵检测等来维护信息系统的安全；有的是依据评估标准而开发的，如 Cobra。不可否认，这些工具的使用会丰富评估所需的系统脆弱、威胁信息、简化评估的工作量，减少评估过程中的主观性，但无论这些工具的功能多么强大，由于信息系统风险评估的复杂性，它在信息系统的风险评估过程中也只能作为辅助手段，代替不了整个风险评估过程。

9.4 风险评估的过程

风险评估过程就是在评估标准的指导下，综合利用相关评估技术、评估方法、评估工具，针对信息系统展开全方位评估工作的完整历程。风险评估在具体实施中一般包括风险评估的准备活动，对信息系统资产、面临威胁、存在的脆弱性的识别，对已采取安全措施的确认，对可能存在的信息安全风险的识别等环节，如图 9.1 所示，下面对各具体步骤进行详细介绍。

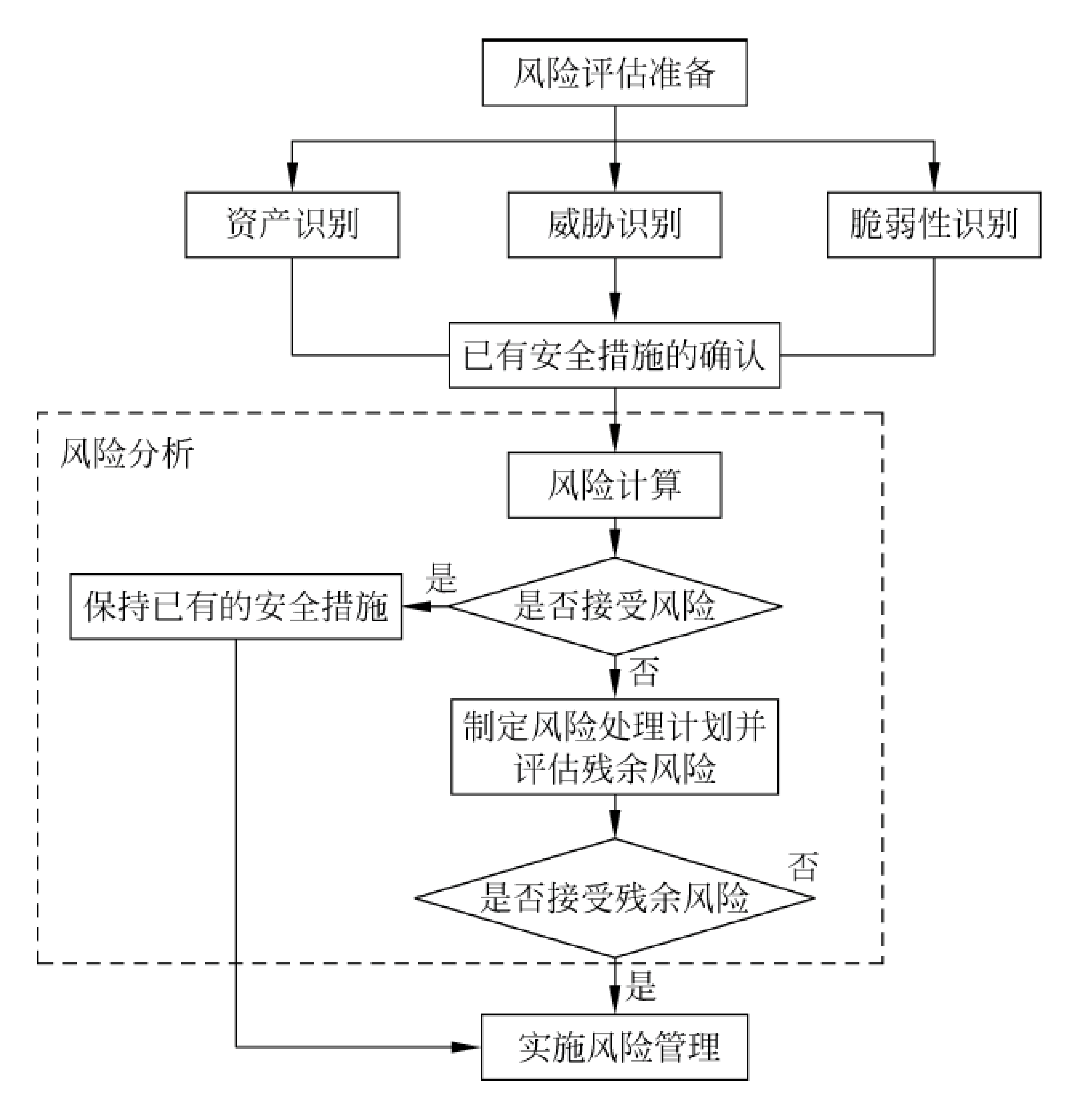


图 9.1 风险评估的过程

9.4.1 风险评估的准备

《信息安全评估规范》指出：“组织实施风险评估是一种战略性的考虑，其结果将受到组织业务战略、业务流程、安全需求、系统规模和机构等方面的影响”。信息安全风险评估不只

是单纯针对信息系统或信息资产本身，它是通过一种系统化的方式，综合分析资产、威胁、脆弱性以及已有安全控制措施之间的内在关系，探求信息资产的安全属性丧失后，对被评估组织（关键业务、外部声誉等方面）所造成的影响。

信息安全风险评估是一项复杂的、系统化的活动，为了保证评估过程的可控性以及评估结果的客观性，在风险评估实施前应进行充分准备和准确计划。按照《信息安全风险评估规范》，评估准备阶段至少包括以下活动。

1. 确定风险评估的目标

信息安全需求是一个组织为保证其业务正常、有效运转而必须达到的信息安全要求，通过分析组织必须符合的相关法律法规、组织在业务流程中对信息安全的机密性、完整性、可用性等方面的需求，来确定风险评估的目标。

2. 确定风险评估的范围

风险评估的范围可能是组织全部的信息及与信息处理相关的各类资产、管理机构，也可能是某个独立的信息系统、关键业务流程、与客户知识产权相关的系统或部门等。

实施一次风险评估的范围可大可小，需要根据具体评估需求确定，可以对组织全部的信息系统进行评估，也可以仅对关键业务流程进行评估，也可以对组织关键部门的信息系统进行评估。

3. 组建评估管理团队和评估实施团队

在确定风险评估的目标、范围后，需要组建风险评估团队，具体执行组织的风险评估。由于风险评估涉及组织管理、业务、信息资产等各个方面，因此风险评估团队中除了信息安全风险评估专业人员外，还需要有组织管理层、相关业务骨干、信息安全运营管理人员等的参与，以便更好地了解组织信息安全状况，以利于风险评估的实施。必要时，可组建由评估方、被评估方领导和相关部门负责人参加的风险评估领导小组，聘请相关专业的技术专家和技术骨干组成专家小组。

评估实施团队应做好评估前的表格、文档、检测工具等各项准备工作，进行风险评估技术培训和保密教育，制定风险评估过程管理相关规定。可根据被评估方要求，双方签署保密合同，必要时签署个人保密协议。

4. 进行系统调研

在确定了风险评估的目标、范围、团队后，要进行系统调研，系统调研是确定被评估对象的过程，风险评估小组应进行充分的系统调研，为风险评估依据和方法的选择、评估内容的实施奠定基础。系统调研内容包括：

（1）组织业务战略，即主要业务职能及未来发展规划；

（2）组织管理制度；

（3）主要业务功能和要求；

（4）网络结构与网络环境，包括内部连接和外部连接；

（5）系统边界；

（6）主要硬件、软件；

（7）数据和信息；

（8）系统和数据的敏感性；

(9) 系统使用人员;

(10) 其他。

系统调研可以采用问卷调查、现场访谈等方法进行。

5. 确定评估标准和方法

项目实施过程中,应依据现有国际或国家信息安全标准,保证评估的规范性。同时参考相关的行业标准或组织自身的策略,增强风险评估的针对性。

根据评估依据,并综合考虑评估的目的、范围、时间、效果、评估人员素质等因素,选择具体的风险计算方法,并依据组织业务实施对系统安全运行的需求,确定相关的评估判断依据,使之能够与组织环境和安全要求相适应。

6. 获得最高管理者对风险评估工作的支持

就上述内容形成较为完整的风险评估实施方案,并报组织最高管理者批准,以获得其对风险评估方案的支持,同时在组织范围内就风险评估相关内容对管理者和技术人员进行培训,以明确有关人员在风险评估中的任务。

9.4.2 资产识别

资产(Asset)是指对组织具有价值的信息或资源,是安全策略保护的对象。安全评估需要确定信息系统的资产,并明确资产的价值,因为价值不同将导致风险值不同,资产的价值不是以资产的经济价值来衡量的,而是由资产在机密性、完整性和可用性这三个安全属性上达成的程度或者其安全属性未达成时所造成的影响程度来决定的。安全属性达成的程度的不同将使资产具有不同的价值,而资产面临的威胁、存在的脆弱性,以及已采用的安全措施都将对资产安全属性的达成程度产生影响。为此,应对组织中的资产进行识别。

资产的范围很广,一切需要加以保护的东西都算作资产,包括信息资产、纸质文件、软件资产、物理资产、人员、公司形象和声誉、服务等。资产的评估应当从关键业务开始,最终覆盖所有的关键资产。对于提供多种业务的组织,其支持业务持续运行的系统数量可能很多,首先需要将信息系统中的资产进行恰当的分类。

1. 资产分类

在实际工作中,具体的资产分类方法可以根据具体的评估对象和要求,由评估者灵活把握。根据资产的表现形式,可将资产分为数据、软件、硬件、文档、服务、人员等类型。表 9.1 列出了一种资产分类的方法。

表 9.1 一种基于表现形式的资产分类方法

分类	示 例
数据	保存在信息媒介上的各种数据资料,包括源代码、数据库数据、系统文档、运行管理规范、计划、报告、用户手册、各类纸质文档等
软件	系统软件:操作系统、数据库管理系统、语言包、工具软件、各种库等 应用软件:办公软件、各类工具软件等 源程序:各种共享源代码、自行或合作开发的各种代码等

续表

分类	示　例
硬件	网络设备：路由器、网关、交换机等 计算机设备：大型机、小型机、服务器、工作站、台式计算机、便携计算机等 存储设备：磁带机、磁盘阵列、磁带、光盘、软盘、移动硬盘等 传输线路：光纤、双绞线等 保障设备：UPS、变电设备、空调、保险柜、文件柜、门禁、消防设施等 安全设备：防火墙、入侵检测系统、身份鉴别等 其他：打印机、复印机、扫描仪、传真机等
服务	办公服务：为提高效率而开发的管理信息系统，包括各种内部配置管理、文件流转管理等服务 网络服务：各种网络设备、设施提供的网络连接服务 信息服务：对外依赖该系统开展的各类服务
文档	纸质的各种文件，如传真、电报、财务报告、发展计划等
人员	掌握重要信息和核心业务的人员，如主机维护主管、网络维护主管及应用项目经理等
其他	企业形象、客户关系等

2. 资产赋值

对资产的赋值不仅要考虑资产的经济价值，更重要的是要考虑资产的安全状况，即资产的机密性、完整性和可用性，对组织信息安全性的影响程度。举个例子来说，美国微软公司若丢失了一台存有最新版本 Windows 操作系统源代码的笔记本电脑，这个电脑丢失事件的发生对微软公司业务造成的损失要比资产本身(笔记本电脑)的价值要大得多。资产赋值的过程也就是对资产在机密性、完整性和可用性上的要求进行分析，并在此基础上得出综合结果的过程。资产对机密性、完整性和可用性上的要求可由安全属性缺失时造成的影响来表示，这种影响可能造成某些资产的损害以致危及信息系统，还可能导致经济利益、市场份额、组织形象的损失。

1）机密性赋值

根据资产在机密性上的不同要求，将其分为 5 个不同的等级，分别对应资产在机密性缺失时对整个组织的影响，表 9.2 提供了一种机密性赋值的参考。

表 9.2　资产保密性赋值表

赋值	标识	定　义
5	很高	包含组织最重要的秘密，关系未来发展的前途命运，对组织根本利益有着决定性的影响，如果泄露会造成灾难性的损害
4	高	包含组织的重要秘密，如果泄露会使组织的安全和利益遭受严重损害
3	中等	包含组织的一般性秘密，如果泄露会使组织的安全和利益受到损害
2	低	仅能在组织内部或组织某一部门内部公开的信息，向外扩散有可能对组织的利益造成轻微损害
1	很低	可对社会公开的信息、公用的信息处理设备和系统资源

2）完整性赋值

根据资产在完整性上的不同要求，将其分为 5 个不同的等级，分别对应资产在完整性缺失时对整个组织的影响，表 9.3 提供了一种完整性赋值的参考。

表 9.3 资产完整性赋值表

赋值	标识	定义
5	很高	完整性价值非常关键，未经授权的修改或破坏会对组织造成重大的或无法接受的影响，对业务冲击很大，并可能造成严重的业务中断，损失难以弥补
4	高	完整性价值较高，未经授权的修改或破坏会对组织造成重大影响，对业务冲击严重，损失较难弥补
3	中等	完整性价值中等，未经授权的修改或破坏会对组织造成影响，对业务冲击明显，但损失可以弥补
2	低	完整性价值较低，未经授权的修改或破坏会对组织造成轻微影响，对业务冲击轻微，损失容易弥补
1	很低	完整性价值非常低，未经授权的修改或破坏会对组织造成的影响可以忽略，对业务冲击可以忽略

3）可用性赋值表

根据资产在可用性上的不同要求，将其分为 5 级，分别对应资产在可用性上缺失时对整个组织的影响。表 9.4 提供了一种可用性赋值的参考。

表 9.4 资产可用性赋值表

赋值	标识	定义
5	很高	可用性价值非常高，合法使用者对信息及信息系统的可用度达到年度 99.9%以上，或系统不允许中断
4	高	可用性价值较高，合法使用者对信息及信息系统的可用度达到每天 90%以上，或系统允许中断时间小于 10min
3	中等	可用性价值中等，合法使用者对信息及信息系统的可用度在正常工作时间达到 70%以上，或系统允许中断时间小于 30min
2	低	可用性价值较低，合法使用者对信息及信息系统的可用度在正常工作时间达到 25%以上，或系统允许中断时间小于 60min
1	很低	可用性价值可以忽略，合法使用者对信息及信息系统的可用度在正常工作时间低于 25%

4）资产重要性等级

资产价值应根据资产在机密性、完整性和可用性上的赋值等级，经过综合评定得出。综合评定方法可以选择对资产机密性、完整性和可用性最为重要的一个属性的赋值等级作为资产的最终赋值结果，也可以根据资产机密性、完整性和可用性赋值进行加权计算得到资产的最终赋值结果。加权方法可根据组织的业务特点确定，在不同的行业中，因为业务、职能和行业背景千差万别，信息安全的目标和安全保障的要求也截然不同，例如电信运营商最关注可用性、金融行业最关注完整性、政府涉密部门最关注保密性，这时机密性、完整性和可用性三者的权值就会相差很大。

为与上述安全属性的赋值相对应，根据最终赋值将资产划分为 5 级，级别越高标识资产越重要，也可以根据组织的实际情况确定资产识别中的赋值依据和等级。表 9.5 中的资产等级划分表明了不同等级的重要性的综合描述。评估者可根据资产赋值结果，确定重要资产的范围，并围绕重要资产进行下一步的风险评估。

表 9.5　资产重要性等级

赋值	标识	定　　义
5	很高	非常重要，其安全属性破坏后可能会对组织造成非常严重的损失
4	高	重要，其安全属性破坏后可能会对组织造成比较严重的损失
3	中等	比较重要，其安全属性破坏后可能会对组织造成中等严重的损失
2	低	不太重要，其安全属性破坏后可能会对组织造成较低的损失
1	很低	不重要，其安全属性破坏后可能会对组织造成很小的损失，甚至可以忽略不计

9.4.3　威胁识别

1. 威胁分类

威胁评估是对信息资产有可能受到的危害进行分析，一般可从威胁来源、威胁途径、威胁意图、损失等几个方面来分析。威胁可能源于对信息系统直接或间接的攻击，例如非授权的泄露、篡改、删除等，也可能源于偶发的或蓄意的事件。一般来说，威胁总是要利用网络中的系统、应用或服务的弱点才可能成功地对资产造成伤害。造成威胁的因素可分为人为因素和环境因素。根据威胁的动机，人为因素又可分为恶意和非恶意两种。环境因素包括自然界不可抗力的因素和其他物理因素。

威胁作用形式可以是对信息系统直接或间接的攻击，也可能是偶发的或蓄意的安全事件，都会在信息的机密性、完整性或可用性等方面造成损害。对威胁进行分类的方式有多种，可以根据其来源、表现形式等对威胁进行分类，表 9.6 提供了一种基于表现形式的威胁分类方法。

表 9.6　威 胁 分 类

种　类	描　　述	威 胁 子 类
软硬件故障	由于设备硬件故障、通信链路中断、系统本身或软件缺陷造成对业务实施、系统稳定运行的影响	设备硬件故障、传输设备故障、存储设备故障、系统软件故障、应用软件故障、数据库软件故障、开发环境故障
物理环境影响	对信息系统正常运行造成影响的物理环境问题和自然灾害	断电、静电、灰尘、湿潮、温度、鼠蚁虫害、电磁干扰、洪灾、火灾、地震
无作为或操作失误	由于应该执行而没有执行相应的操作，或无意地执行了错误的操作，对系统造成的影响	维护错误、操作失误
管理不到位	安全管理措施没有落实，造成安全管理不规范，或者管理混乱，从而破坏信息系统正常有序进行	管理制度和策略不完善、管理规程缺失、职责不明确、监督控管机制不健全等
恶意代码	故意在计算机上执行恶意任务的程序代码	网络病毒、间谍软件、窃听软件、蠕虫等
越权或滥用	通过采用一些措施，超越自己的权限访问资源，或者滥用职权做出破坏信息系统的行为	非授权访问网络资源、非授权访问系统资源、滥用权限非正常修改系统配置或数据、滥用权限泄露秘密信息
网络攻击	利用工具和技术，对信息系统进行攻击和入侵	网络探测和信息采集、漏洞探测、嗅探(账号、口令、权限等)、用户身份伪造和欺骗、用户或业务数据的窃取和破坏、系统运行的控制和破坏等

续表

种类	描　　述	威胁子类
物理攻击	通过物理的接触造成对软件、硬件、数据的破坏	物理接触、物理破坏、窃取等
泄密	信息泄露给不应了解的他人	内部信息泄露、外部信息泄露等
篡改	非法修改信息，破坏信息的完整性使系统的安全性降低或信息不可用	篡改网络配置信息、篡改系统配置信息、篡改安全配置信息、篡改用户身份信息或业务数据信息等
抵赖	不承认收到的信息和所作的操作和交易	原发抵赖、接收抵赖、第三方抵赖等

2. 威胁赋值

威胁出现的频率是衡量威胁严重程度的重要因素，因此威胁识别后需要对威胁频率进行赋值，以带入最后的风险计算中。

评估者应根据经验和有关统计数据来对威胁频率进行赋值，威胁赋值中需要综合考虑以下三方面因素，以形成在某种评估环境中各种威胁出现的频率。

(1) 以往安全事件报告中出现的威胁及其频率的统计；

(2) 实际环境中通过检测工具以及各种日志发现的威胁及其频率的统计；

(3) 近一两年来国际组织发布的对于整个社会或特定行业的威胁及其频率统计，以及发布的威胁预警。

可以对威胁出现的频率进行等级化处理，不同等级分别代表威胁出现的频率的高低。等级数值越大，威胁出现的频率越高。

表 9.7 提供了威胁出现频率的一种赋值方法。在实际的评估中，威胁频率的判断应根据历史统计或行业判断，在评估准备阶段确定，并得到被评估方的认可。

表 9.7　威胁赋值

赋值	标识	定　　义
5	很高	出现的频率很高(或≥1 次/周)；或在大多数情况下几乎不可避免；或可以证实经常发生
4	高	出现的频率较高(或≥1 次/月)；或在大多数情况下很有可能发生；或可以证实多次发生
3	中等	出现的频率较中等(或≥1 次/半年)；或在某种情况下可能发生；或被证实曾经发生
2	低	出现的频率较小；或一般不太可能发生；或没有被证实发生过
1	很低	威胁几乎不可能发生，仅可能在非常罕见和例外的情况下发生

9.4.4 脆弱性识别

脆弱性评估是指通过各种测试方法，获得信息资产中所存在的缺陷清单。脆弱性是资产本身存在的，如果没有被相应的威胁利用，单纯的脆弱性本身不会对资产造成损害。而且如果系统足够强健，即使是严重的威胁也不会导致安全事件发生，即威胁总是利用资产的脆弱性才可能造成危害。

资产的脆弱性具有隐蔽性，有些脆弱性只有在一定的条件和环境下才能显现，这是脆弱

性识别中最为困难的部分。脆弱性识别是风险评估中最重要的一个环节。脆弱性识别可以以资产为核心,针对每一项需要保护的资产,识别可能被威胁利用的弱点,并对脆弱性的严重程度进行评估;也可以从物理、网络、系统、应用等层次进行识别,然后与资产、威胁对应起来。脆弱性识别的依据可以是国际或国家安全标准,也可以是行业规范、应用流程的安全要求。对应用在不同环境中的相同的弱点,其脆弱性严重程度是不同的,评估者应从组织安全策略的角度考虑、判断资产的脆弱性及其严重程度。信息系统所采用的协议、应用流程的完备与否、与其他网络的互连等也应考虑在内。

脆弱性识别的数据应来自于资产的所有者、使用者,以及相关业务领域和软硬件方面的专业人员等。脆弱性识别所采用的方法主要有问卷调查、工具检测、人工核查、文档查阅、渗透测试等,其中渗透测试是一种从攻击者的角度来对主机系统的安全程度进行安全评估的手段,在对现有信息系统不造成任何损害的前提下,模拟入侵者对指定系统进行攻击检测。渗透测试通常能以非常明显、直观的结果来反映出系统的安全现状。可根据具体的评估对象、评估目的来选择脆弱点识别方法。

脆弱性识别主要从技术和管理两个方面进行,技术脆弱性涉及物理层、网络层、系统层、应用层等各个层面的安全问题。管理脆弱性又分为技术管理脆弱性和组织管理脆弱性两方面,前者与具体技术活动相关,后者与管理环境相关。

对不同的识别对象,其脆弱性识别的具体要求应参照相应的技术或管理标准实施。例如,对物理环境的脆弱性识别应按 GB/T 9361 中的技术指标实施;对操作系统、数据库的脆弱性识别应按 GB 17859-1999 中的技术指标实施;对网络、系统、应用等信息技术安全性的脆弱性识别应按 GB/T 18336-2001 中的技术指标实施;对管理的脆弱性识别应按 GB/T 19716-2005 的要求对安全管理制度及其执行情况进行检查,发现管理脆弱性和不足。表 9.8 提供了一种脆弱性识别内容的参考。

表 9.8　脆弱性识别内容表

类　　型	识别对象	识 别 内 容
技术脆弱性	物理环境	从机房场地、防火、供配电、防静电、接地与防雷、电磁防护、通信线路的保护、机房区域防护、机房设备管理等方面进行识别
	网络结构	从网络结构设计、边界防护、外部访问控制策略、内部访问控制策略、网络设备安全配置等方面进行识别
	系统软件	从补丁安装、用户账号、口令策略、资源共享、事件审计、访问控制、新系统配置、注册表加固、网络安全、系统管理等方面进行识别
	数据库软件	从补丁安装、鉴别机制、口令机制、访问控制、网络和服务设置、备份和恢复机制、审计机制等方面进行识别
	应用中间件	从协议安全、交易完整性、数据完整性、通信、鉴别机制、密码保护等方面进行识别
	应用系统	从审计机制、审计存储、访问控制策略、数据完整性、通信、鉴别机制、密码保护等方面进行识别
管理脆弱性	技术管理	从物理和环境安全、通信与操作管理、访问控制、系统开发与维护、业务连续性等方面进行识别
	组织管理	从安全策略、组织安全、资产分类与控制、人员安全、符合性等方面进行识别

可以根据脆弱性对资产的暴露程度、技术实现的难易程度、流行程度等，采用等级方式对已识别的脆弱性的严重程度进行赋值。由于很多弱点反映的是同一方面的问题，或可能造成相似的后果，赋值时应综合考虑这些弱点，以确定这一方面脆弱性的严重程度。

对某个资产，其脆弱性的严重程度还受到组织管理脆弱性的影响。因此，资产的脆弱性赋值还应参考技术管理和组织管理脆弱性的严重程度。

脆弱性严重程度可以进行等级化处理，不同的等级分别代表资产脆弱性严重程度的高低。等级数值越大，脆弱性严重程度越高。表 9.9 提供了脆弱性严重程度的一种赋值方法。

表 9.9 脆弱性严重程度赋值表

赋值	标识	定 义
5	很高	如果被威胁利用，将对资产造成完全损害
4	高	如果被威胁利用，将对资产造成重大损害
3	中等	如果被威胁利用，将对资产造成一般损害
2	低	如果被威胁利用，将对资产造成较小损害
1	很低	如果被威胁利用，将对资产造成的损害可以忽略

9.4.5 已有安全措施确认

在识别脆弱性的同时，评估人员应对已采取的安全措施的有效性进行确认。安全措施的确认应评估其有效性，即是否真正地降低了系统的脆弱性，抵御了威胁。对有效的安全措施继续保持，以避免不必要的工作和费用，防止安全措施的重复实施。对确认为不适当的安全措施应该核实是否应被取消或对其进行修正，或用更合适的安全措施替代。

安全措施可以分为预防性安全措施和保护性安全措施两种。预防性安全措施可以降低威胁利用脆弱性导致安全事件发生的可能性，如入侵检测系统；保护性安全措施可以减少因安全事件发生而对组织或系统造成的影响。

已有安全措施确认与脆弱性识别存在一定的联系。一般来说，安全措施的使用将减少系统技术或管理上的脆弱性，但安全措施确认并不需要像脆弱性识别过程那样具体到每个资产、组件的脆弱性，而是一类具体措施的集合，为风险处理计划的制定提供依据与参考。

9.4.6 风险分析

风险分析的原理如图 9.2 所示，风险分析的主要内容如下。

(1) 根据威胁及威胁利用弱点的难易程度判断安全事件发生的可能性；

(2) 根据脆弱性的严重程度及安全事件所作用资产的价值计算安全事件的损失；

(3) 根据安全事件发生的可能性以及安全事件的损失，计算安全事件一旦发生对组织的影响，即风险值。

1. 风险计算原理

在完成了资产识别、威胁识别、脆弱性识别，以及已有安全措施确认后，将采用适当的方法和工具确定威胁利用脆弱性导致安全事件发生的可能性。综合安全事件所作用的资产价值及脆弱性的严重程度，判断安全事件造成的损失对组织的影响，即安全风险。下面使用范

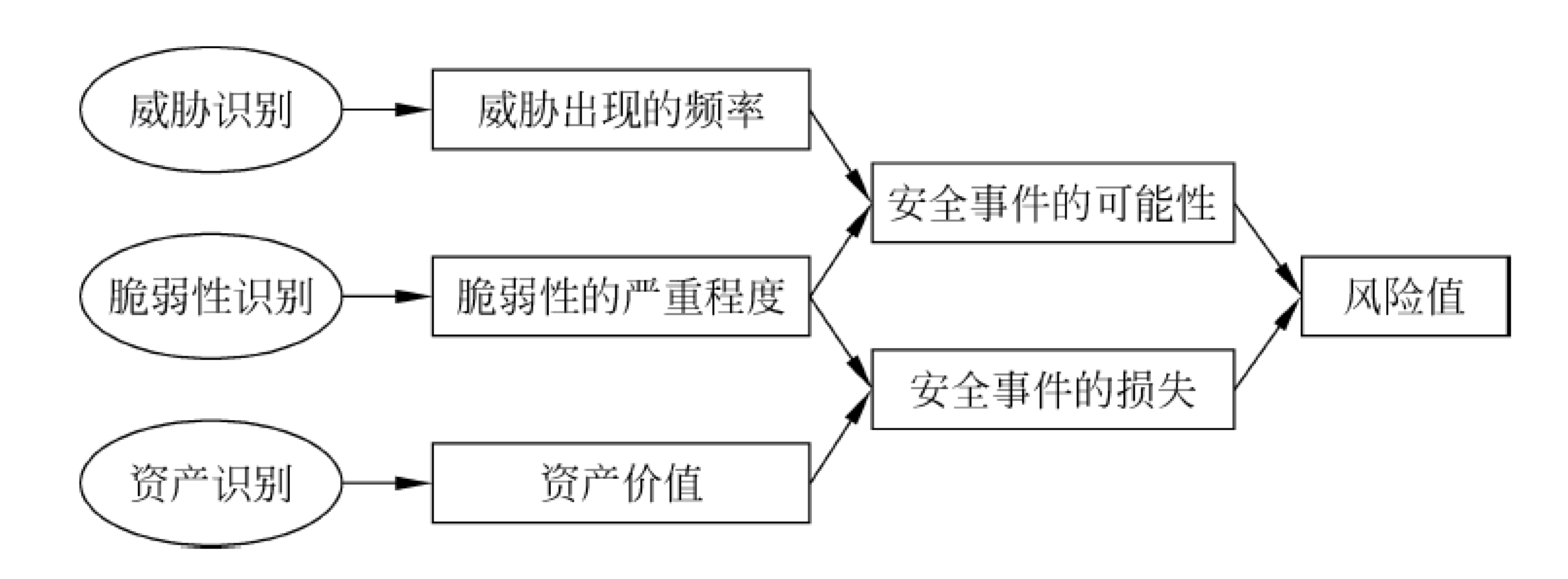


图 9.2　风险分析原理图

式来形式化说明风险计算原理：

$$风险值=R(A,T,V)=R(L(T,V),F(I_a,V_a))$$

其中：R 表示安全风险计算函数，A 表示资产，T 表示威胁出现频率，V 表示脆弱性，I_a 表示安全事件所作用的资产价值，V_a 表示脆弱性严重程度，L 表示威胁利用资产脆弱性导致安全事件发生的可能性，F 表示安全事件发生后产生的损失。

在风险计算中有以下三个关键计算环节。

1）计算安全事件发生的可能性

根据威胁出现频率和脆弱性的状况，计算威胁利用脆弱性导致安全事件发生的可能性，即：

$$安全事件发生的可能性=L(威胁出现频率,脆弱性)=L(T,V)$$

在具体评估中，应综合攻击者技术能力（专业技术程度、攻击设备等）、脆弱性被利用的难易程度（可访问时间、设计和操作知识公开程度等）、资产吸引力等因素来判断安全事件发生的可能性。

2）计算安全事件发生后的损失

根据资产价值及脆弱性严重程度，计算安全事件一旦发生后的损失，即：

$$安全事件的损失=F(资产价值,脆弱性严重程度)=F(I_a,V_a)$$

部分安全事件的发生造成的损失不仅仅针对该资产本身，还可能影响业务的连续性；不同安全事件的发生对组织造成的影响也是不一样的。在计算某个安全事件的损失时，应将对组织的影响考虑在内。

部分安全事件造成的损失的判断还应参照安全事件发生可能性的结果，对发生可能性极小的安全事件，如处于非地震带的地震威胁、在采取完备供电措施状况下的电力故障威胁等，可以不计算其损失。

3）计算风险值

根据计算出的安全事件发生的可能性以及安全事件造成的损失计算风险值，即：

$$\begin{aligned}风险值&=R(安全事件发生的可能性,安全事件的损失)\\&=R(L(T,V),F(I_a,V_a))\end{aligned}$$

评估者可根据自身情况选择相应的风险计算方法计算风险值，如矩阵法或相乘法。矩阵法通过构造一个二维矩阵，形成安全事件发生的可能性与安全事件的损失之间的二维关系；相乘法通过构造经验函数，将安全事件发生的可能性与安全事件的损失进行运算得到风险值。

目前常用的风险值计算方法有矩阵法和相乘法，这里简要介绍相乘法。

2. 使用相乘法计算风险

1）相乘法的原理

相乘法主要用于两个或多个要素值确定一个要素值的情形。即 $z=f(x,y)$，函数 f 可以采用相乘法。相乘法的原理是：

$z=f(x,y)=x\odot y$。

当 f 为增量函数时，$\odot$ 可以为直接相乘，也可以为相乘后取模等，例如：

$z=f(x,y)=x\times y$，或 $z=f(x,y)=\sqrt{(x\times y)}$ 等。

相乘法提供一种定量的计算方法，直接使用两个要素值进行相乘得到另一个要素的值。相乘法的特点是简单明确，直接按照统一公式计算，即可得到所需结果。

在风险值计算中，通常需要对两个要素确定的另一个要素值进行计算，例如由威胁和脆弱性确定安全事件发生的可能性值、由资产和脆弱性确定安全事件的损失值，因此相乘法在风险分析中得到广泛采用。

2）计算示例

假设某信息系统共有两个重要资产：资产 A1 和资产 A2；

资产 A1 面临三个主要威胁：威胁 T1、威胁 T2 和威胁 T3；

资产 A2 面临两个主要威胁：威胁 T4 和威胁 T5；

威胁 T1 可以利用的资产 A1 存在的一个脆弱性 V1；

威胁 T2 可以利用的资产 A1 存在的两个脆弱性 V2、V3；

威胁 T3 可以利用的资产 A1 存在的一个脆弱性 V4；

威胁 T4 可以利用的资产 A2 存在的一个脆弱性 V5；

威胁 T5 可以利用的资产 A2 存在的一个脆弱性 V6；

资产价值分别是：资产 A1=4，资产 A2=5；

威胁发生频率分别是：威胁 T1=1，威胁 T2=5，威胁 T3=4，威胁 T4=3，威胁 T5=4；

脆弱性严重程度分别是：脆弱性 V1=3，V2=1，V3=5，V4=4，V5=4，V6=3。

两个资产的风险值计算过程类似，下面以资产 A1 为例使用相乘法计算风险值。

资产 A1 面临的主要威胁包括威胁 T1、威胁 T2 和威胁 T3，威胁 T1 可以利用的资产 A1 存在的脆弱性有一个，威胁 T2 可以利用的资产 A1 存在的脆弱性有两个，威胁 T3 可以利用的资产 A1 存在的脆弱性有一个，则资产 A1 存在的风险值有 4 个。4 个风险值的计算过程类似，下面以资产 A1 面临的威胁 T1 可以利用的脆弱性 V1 为例，计算安全风险值。其中计算公式使用：

$z=f(x,y)=\sqrt{(x\times y)}$，并对 z 的计算值四舍五入取整得到最终结果。

（1）计算安全事件发生的可能性

威胁发生频率：威胁 T1=1；

脆弱性严重程度：脆弱性 V1=3。

计算安全事件发生的可能性，安全事件发生的可能性 $=\sqrt{(1\times 3)}=\sqrt{3}$。

（2）计算安全事件的损失

资产价值：资产 A1=4；

脆弱性严重程度：脆弱性 V1=3。

计算安全事件的损失，安全事件损失＝$\sqrt{(4\times3)}=\sqrt{12}$。

(3) 计算风险值

安全事件发生的可能性＝$\sqrt{3}$；

安全事件损失＝$\sqrt{12}$。

安全事件风险值＝$\sqrt{3}\times\sqrt{12}=6$。

按照上述方法进行计算，得到资产 A1 的其他风险值，以及资产 A2 和资产 A3 的风险值，然后进行风险结果等级判定。

3) 结果判定

确定风险等级划分如表 9.10 所示。

表 9.10　风险等级划分

风险值	1～5	6～10	11～15	16～20	21～25
风险等级	1	2	3	4	5

根据上述计算方法，以此类推，得到两个重要资产的风险值，并根据风险等级划分表，确定风险等级，结果如表 9.11 所示。

表 9.11　风险等级

资产	威胁	脆弱性	风险值	风险等级
资产 A1	威胁 T1	脆弱性 V1	6	2
	威胁 T2	脆弱性 V2	4	1
	威胁 T2	脆弱性 V3	22	5
	威胁 T3	脆弱性 V4	16	4
资产 A2	威胁 T4	脆弱性 V5	15	3
	威胁 T5	脆弱性 V6	13	3

3. 风险结果判定

为实现对风险的控制和管理，可以对风险评估的结果进行等级化处理。可以将风险划分为一定的级别，等级越高，风险越大。

评估者应根据所采用的风险计算方法，计算每种资产面临的风险值，根据风险值的分布状况，为每个等级设定风险值范围，并对所有风险计算结果进行等级处理。每个等级代表了相应风险的严重程度。表 9.12 提供了一种风险评估等级划分方法。

表 9.12　风险评估等级划分表

赋值	标识	定　　义
5	很高	一旦发生将产生非常严重的经济或社会影响，如组织信誉严重破坏、严重影响组织的正常经营，经济损失重大、社会影响恶劣
4	高	一旦发生将产生较大的经济或社会影响，在一定的范围内给组织的经营和组织信誉造成损害
3	中等	一旦发生会造成一定的经济、社会或生产经营影响，但影响面和影响程度不大
2	低	一旦发生造成的影响程度较低，一般仅限于组织内部，通过一定手段很快能解决
1	很低	一旦发生造成的影响几乎不存在，通过简单的措施就能弥补

风险等级划分是为了在风险管理过程中对不同风险进行直观的比较,以确定组织安全策略。组织应当综合考虑风险控制成本与风险造成的影响,提出一个可接受的风险范围。对某些资产的风险,如果风险计算值在可接受的范围内,则该风险是可接受的风险,应保持已有的安全措施;如果风险评估值在可接受的范围外,即风险计算值高于可接受范围的上限值,是不可接受的风险,需要采取安全措施以降低、控制风险。

9.4.7 风险处理计划

对不可接受的风险应根据导致该风险的脆弱性制订风险处理计划。风险处理计划中应明确采取的弥补脆弱性的安全措施、预期效果、实施条件、进度安排、责任部门等。安全措施的选择应从管理和技术两个方面考虑。安全措施的选择与实施应参考信息安全的相关标准进行。

9.4.8 残余风险的评估

在对不可接受的风险选择适当安全措施后,为确保安全措施的有效性,可进行再评估,以判断实施安全措施后的残余风险是否已经降低到可接受的水平。残余风险的评估可以依据本标准提出的风险评估流程实施,也可作适当裁剪。一般来说,安全措施的实施是以减少脆弱性或降低安全事件发生可能性为目标的,因此,残余风险的评估可以从脆弱性评估开始,在对照安全措施实施前后的脆弱性状况后,再次计算风险值的大小。

某些风险可能在选择了适当的安全措施后,残余风险的结果仍处于不可接受的风险范围内,应考虑是否接受此风险或进一步增加相应的安全措施。

9.4.9 风险评估文档

风险评估文档是指在整个风险评估过程中产生的评估过程文档和评估结果文档,包括(但不仅限于此)以下几种。

(1) 风险评估方案:阐述风险评估的目标、范围、人员、评估方法、评估结果的形式和实施进度等。

(2) 风险评估程序:明确评估的目的、职责、过程、相关的文档要求,以及实施本次评估所需要的各种资产、威胁、脆弱性识别和判断依据。

(3) 资产识别清单:根据组织在风险评估程序文档中所确定的资产分类方法进行资产识别,形成资产识别清单,明确资产的责任人/部门。

(4) 重要资产清单:根据资产识别和赋值的结果,形成重要资产列表,包括重要资产名称、描述、类型、重要程度、责任人/部门等。

(5) 威胁列表:根据威胁识别和赋值的结果,形成威胁列表,包括威胁名称、种类、来源、动机及出现的频率等。

(6) 脆弱性列表:根据脆弱性识别和赋值的结果,形成脆弱性列表,包括具体脆弱性的名称、描述、类型及严重程度等。

(7) 已有安全措施确认表:根据对已采取的安全措施确认的结果,形成已有安全措施确认表,包括已有安全措施名称、类型、功能描述及实施效果等。

(8) 风险评估报告:对整个风险评估过程和结果进行总结,详细说明被评估对象、风险评估方法、资产、威胁、脆弱性的识别结果、风险分析、风险统计和结论等内容。

(9) 风险处理计划：根据风险评估程序，要求风险评估过程中的各种现场记录可复现评估过程，并作为产生歧义后解决问题的依据。

对于风险评估过程中形成的相关文档，还应规定其标识、存储、保护、检索、保存期限以及处置所需的控制。相关文档是否需要以及详略程度由组织的管理者来决定。

9.5　信息系统风险评估发展存在的问题

目前"信息系统安全是一项系统工程"的观点已得到广泛的认可、接受，作为该工程的基础和前提的风险评估也越来越受到大家的重视，但在该领域的研究、发展过程中还需要纠正和解决一些模糊概念和问题。

第一，安全评估体系所应包括的相应组织架构、业务、标准和技术体系还不完善。

第二，不能简单地将系统风险评估理解为一个具体的产品、工具，系统的风险评估更应该是一个过程，是一个体系。完善的系统风险评估体系应包括相应的组织架构、业务体系、标准体系和技术体系。

第三，在评估标准的采用上，没有统一的标准，由于各种标准的侧重点不同，导致评估结果没有可比性，甚至会出现较大的差异，而且目前国内还缺乏具有自主知识产权、比较系统的信息系统评估标准。

第四，评估过程的主观性也是影响评估结果的一个相当重要而又最难解决的方面，在信息系统风险评估中，主观性是不可避免的，我们所要做的是尽量减少人为主观性，目前在该领域利用神经网络、专家系统、分类树等人工智能技术进行的研究比较活跃。

第五，风险评估工具比较缺乏，市场上漏洞扫描、防火墙等都有比较成熟的产品，但与信息系统风险评估相关的工具却很匮乏。

9.6　小　　结

信息安全风险评估是信息安全保障体系建立过程中的重要评价方法和决策机制，没有准确及时的风险评估，将无法对信息安全状况做出准确的判断。安全评估作为信息系统安全工程重要组成部分，已经不仅仅是个别企业的问题，而是关系到国民经济的每一方面的重大问题，它将逐渐走上规范化和法制化的轨道上来，国家对各种配套的安全标准和法规的制定将会更加健全，评估模型、评估方法、评估工具的研究、开发将更加活跃，信息系统及相关产品的风险评估认证将成为必需环节。

习　　题

一、填空题

1. 我国开展信息安全风险评估工作遵循的国家标准是________。
2. 风险评估的方法有________、________和________。

3. 资产赋值的过程也就是对资产在________、________和________的要求进行分析，并在此基础上得出综合结果的过程。

4. 常用的风险值计算方法有________和________。

5. 风险评估的工具可以分为________、________和________。

6. 资产的价值不是以资产的经济价值来衡量的，而是由资产在________、________和________这三个安全属性上的达成程度或者其安全属性未达成时所造成的影响程度来决定的。

7. 威胁识别后需要对________进行赋值。

二、简答题

1. 什么是信息安全风险评估？

2. 风险计算的主要过程包括哪些步骤？

3. 什么是资产，什么是资产价值？

4. 什么是安全威胁？产生安全威胁的主要因素是什么？

5. 什么是脆弱性？什么是脆弱性识别？

6. 威胁的可能性赋值受到哪些因素的影响？

7. 请谈谈计算机信息系统安全风险评估在信息安全建设中的地位和重要意义。

8. 简述在风险评估时从哪些方面收集风险评估的数据。

9. 查阅相关文献资料，了解对信息系统进行安全风险评估的其他方法，比较它们的优缺点，并选择一种评估方法对本单位或个人的信息系统安全做一次风险评估。

10. 知识拓展：查阅中国信息安全风险评估论坛、国家信息中心信息安全风险评估网，了解更多安全风险评估理论和技术的进展。

第 10 章　恶意代码检测与防范技术

恶意代码(Malicious Code)是指在未授权的情况下,以破坏软硬件设备、窃取用户信息、干扰用户正常使用、扰乱用户心理为目的而编制的软件或代码片段。

常见的恶意代码包括计算机病毒、蠕虫、特洛伊木马、后门、RootKit 等,它们以各种方式侵入计算机系统,对信息系统的正常使用造成了极大危害,主要包括攻击系统,造成系统瘫痪或操作异常;危害数据文件的安全存储和使用;泄露隐私信息;肆意占用资源,影响系统或网络的性能;攻击应用程序,如影响邮件的收发等。

本章简要介绍几类恶意代码,其中 10.1 节介绍计算机病毒的结构、原理以及病毒检测的方法,10.2 节介绍特洛伊木马的功能和特点,分析其工作机理,给出木马检测与防范技术,10.3 节介绍蠕虫的定义、基本结构和工作原理以及蠕虫防范的方法。

10.1　计算机病毒

早在 1949 年,计算机先驱 John von Neumann 在论文中指出程序可以被编写成能自我复制并增加自身大小的形式。1977 年 Thomas. J. Ryan 推出了轰动一时的科幻小说 *The Adolescence of P-1*,在这本书中,作者构造了一种神秘的、能自我复制、利用信息通道传播的计算机程序,称为计算机病毒,这些病毒漂泊于计算机内,游荡于硅片之间,控制 7000 多台计算机操作系统,引起混乱和不安。这一科幻故事很快变成了现实。

第一个被检测到的病毒出现在 1986 年,称为 Brain(巴基斯坦智囊病毒),这是一个驻留内存的根扇区病毒。

10.1.1　定义

1983 年 11 月,Fred Cohen 在一次计算机安全学术会议上首次提出计算机病毒的概念:计算机病毒是一个能够通过修改程序,把自身复制进去,进而感染其他程序的程序。这一概念强调了计算机病毒能够“传染”其他程序这一特点。

我国在《中华人民共和国计算机信息系统安全保护条例》中将病毒定义为:编制或者在计算机程序中插入的破坏计算机功能或者数据,影响计算机使用并且能够自我复制的一组计算机指令或者程序代码。为什么把这种恶意代码称为病毒呢,主要的原因是计算机病毒和生物病毒有极为相似的特征,计算机病毒一般具有以下特征。

1. 传染性

与生物界中的病毒可以从一个生物体传播到另一个生物体一样,计算机病毒可以借助各种渠道(移动存储介质、共享目录、邮件等)从已经感染的计算机系统扩散到其他计算机系统。

2. 潜伏性

计算机病毒的潜伏性是指计算机病毒可以依附于其他媒体寄生的能力,侵入后的病毒一般不会马上破坏系统而是潜伏到条件成熟才发作。

3. 触发性

潜伏下来的计算机病毒一般要在一定的条件下才被激活,发起攻击。病毒的触发条件可以是日期/时间触发、键盘触发、计数器触发、启动触发等。

4. 非授权执行

计算机病毒隐蔽在合法程序和数据中,当用户运行正常程序时,病毒伺机取得系统的控制权,先于正常程序执行,并对用户呈透明状态。

5. 破坏性

计算机病毒的破坏性包括以下几方面。

(1) 对计算机数据信息的直接破坏:删除、改名、替换内容、假冒文件、磁盘格式化等。

(2) 抢占系统资源:占用和消耗内存、硬盘等资源。

(3) 影响计算机运行速度。

(4) 对计算机硬件主板、显示器、打印机等的破坏。

按照计算机病毒的特点及特性,计算机病毒的分类方法有许多种。

1. 按照计算机病毒攻击的系统分类

(1) 攻击 Windows 系统的病毒。由于 Windows 的图形用户界面(GUI)和多任务操作系统深受用户的欢迎,因而是病毒攻击的主要对象。首例破坏计算机硬件的 CIH 病毒就是一个 Windows 95/98 病毒。

(2) 攻击 UNIX/Linux 系统的病毒。当前,UNIX/Linux 系统应用非常广泛,许多大型机均采用 UNIX/Linux 作为操作系统,UNIX/Linux 病毒的出现,对人类的信息处理也是一个严重的威胁。

2. 按照计算机病毒的链接方式分类

由于计算机病毒本身必须有一个攻击对象以实现对计算机系统的攻击,计算机病毒所攻击的对象是计算机系统可执行的部分。

1) 源码型病毒

该病毒攻击高级语言编写的程序,该病毒在高级语言所编写的程序编译前插入到源程序中,经编译成为合法程序的一部分。

2) 嵌入型病毒

这种病毒是将自身嵌入到现有程序中,把计算机病毒的主体程序与其攻击的对象以插入的方式链接。这种计算机病毒是难以编写的,一旦侵入程序体后也较难消除。

3) 外壳型病毒

外壳型病毒将其自身包围在主程序的四周,对原来的程序不作修改。这种病毒最为常见,易于编写,也易于发现,一般测试文件的大小即可知。

4) 操作系统型病毒

这种病毒在运行时,用自己的逻辑部分取代操作系统的合法程序模块,具有很强的破坏

力，可以导致整个系统瘫痪。圆点病毒和大麻病毒就是典型的操作系统型病毒。

3. 按照计算机病毒的破坏情况分类

1）良性计算机病毒

良性病毒是指不包含立即对计算机系统产生直接破坏作用的代码的病毒。这类病毒为了表现其存在，只是不停地进行扩散，从一台计算机传染到另一台，并不破坏计算机内的数据。其实良性、恶性都是相对而言的。良性病毒取得系统控制权后，会导致整个系统运行效率降低，系统可用内存总数减少，使某些应用程序不能运行。它还与操作系统和应用程序争抢 CPU 的控制权，导致整个系统死锁，给正常操作带来麻烦。

2）恶性计算机病毒

恶性病毒就是指在其代码中包含损伤和破坏计算机系统的操作，在其传染或发作时会对系统产生直接的破坏作用的病毒。这类病毒是很多的，如米开朗基罗病毒。当米氏病毒发作时，硬盘的前 17 个扇区被彻底破坏，使整个硬盘上的数据无法恢复，造成的损失是无法挽回的。有的病毒还会对硬盘做格式化等破坏，这些操作代码都是刻意编写进病毒的，这是其本性之一。

4. 按照计算机病毒的寄生部位或传染对象分类

传染性是计算机病毒的本质属性，根据寄生部位或传染对象分类，即根据计算机病毒传染方式进行分类，有以下几种。

1）磁盘引导区传染的计算机病毒

磁盘引导区传染的病毒主要是用病毒的全部或部分逻辑取代正常的引导记录，而将正常的引导记录隐藏在磁盘的其他地方。由于引导区是磁盘能正常使用的先决条件，因此，这种病毒在运行的一开始（如系统启动）就能获得控制权，其传染性较大。由于在磁盘的引导区内存储着需要使用的重要信息，如果对磁盘上被移走的正常引导记录不进行保护，则在运行过程中就会导致引导记录的破坏。引导区传染的计算机病毒较多，例如“大麻”和“小球”病毒就是这类病毒。

2）操作系统传染的计算机病毒

操作系统是一个计算机系统得以运行的支持环境，它包括 COM、EXE 等许多可执行程序及程序模块。操作系统传染的计算机病毒就是利用操作系统中所提供的一些程序及程序模块寄生并传染的。通常，这类病毒作为操作系统的一部分，只要计算机开始工作，病毒就处在随时被触发的状态，而操作系统的开放性和不绝对完善性给这类病毒出现的可能性与传染性提供了方便。操作系统传染的病毒目前已广泛存在，“黑色星期五”即为此类病毒。

3）可执行程序传染的计算机病毒

可执行程序传染的病毒通常寄生在可执行程序中，一旦程序被执行，病毒也就被激活，病毒程序首先被执行，并将自身驻留在内存，然后设置触发条件，进行传染。

5. 按照计算机病毒激活的时间分类

按照计算机病毒激活的时间可分为定时病毒和随机病毒。定时病毒仅在某一特定时间才发作，而随机病毒一般不是由时钟来激活的。

6. 按照寄生方式和传染途径分类

计算机病毒按其寄生方式可分为两类，一是引导型病毒，二是文件型病毒；它们再按其

传染途径又可分为驻留内存型和不驻留内存型,驻留内存型按其驻留内存方式又可细分。混合型病毒集引导型和文件型病毒特性于一体。

1）引导型病毒

引导型病毒是一种在 ROM BIOS 之后,系统引导时出现的病毒,它先于操作系统,依托的环境是 BIOS 中断服务程序。引导型病毒常驻在内存中。

引导型病毒按其寄生对象的不同又可分为两类,即 MBR(主引导区)病毒和 BR(引导区)病毒。MBR 病毒也称为分区病毒,将病毒寄生在硬盘分区主引导程序所占据的硬盘 0 头 0 柱面第 1 个扇区中。典型的病毒有大麻(Stoned)、2708 等。BR 病毒是将病毒寄生在硬盘逻辑 0 扇区或软盘逻辑 0 扇区(即 0 面 0 道第 1 个扇区)。典型的病毒有 Brain、小球病毒等。

2）文件型病毒

文件型病毒主要以感染可执行程序为主。它的安装必须借助于病毒的载体程序,即要运行病毒的载体程序,方能把文件型病毒引入内存。感染病毒的文件被执行后,病毒通常会趁机再对下一个文件进行感染。

文件型病毒分为源码型病毒、嵌入型病毒和外壳型病毒。源码型病毒是用高级语言编写的,若不进行汇编、链接则无法传染扩散。嵌入型病毒是嵌入在程序中间的,它只能针对某个具体程序,如 dBASE 病毒。文件外壳型病毒按其驻留内存方式可分为高端驻留型、常规驻留型、内存控制链驻留型、设备程序补丁驻留型和不驻留内存型。

【例 10-1】 CIH 病毒。

产生于 1998 年的 CIH 病毒属于文件型病毒,通过网络或移动介质传播,主要感染 Windows 95/98/Me 系统下的可执行文件,CIH 发作时,可以破坏计算机的主板和硬盘。CIH 病毒会向主板可擦写的 BIOS 中写入垃圾信息,导致 BIOS 中的内容被洗去,造成计算机无法启动。CIH 病毒一般潜伏在系统中,并在固定的日期发作。

10.1.2 计算机病毒的结构

根据计算机病毒的工作流程,病毒一般由感染标记、感染模块、触发模块、破坏模块(表现模块)和主控模块构成。

感染标记又称病毒签名。病毒程序感染宿主程序时,要把感染标记写入宿主程序,作为该程序已被感染的标记。感染标记是一些数字或字符串,通常以 ASCII 方式存放在程序里。病毒在感染健康程序以前,先要对感染对象进行搜索,查看它是否带有感染标记。如果有,说明它被感染过,就不再进行感染;如果没有,病毒就感染该程序。不同的病毒感染标记位置不同,内容不同。例如,巴基斯坦病毒感染标记在 BOOT 扇区的 O4H 处,内容为 1234H;大麻病毒在主引导扇区或 BOOT 扇区的 0H 处,内容为 EA0500C007;耶路撒冷病毒在感染文件的尾部,内容是 MsDos。

感染模块是病毒进行感染时的动作部分,感染模块主要做 3 件事:①寻找一个可执行文件;②检查该文件是否有感染标记;③如果没有感染标记,进行感染,将病毒代码放入宿主程序。

破坏模块负责实现病毒的破坏动作,其内部是实现病毒编写者预定的破坏动作的代码。这些破坏动作可能是破坏文件、数据,破坏计算机的时间效率和空间效率或者使机器崩溃。

触发模块根据预定条件满足与否，控制病毒的感染或破坏动作。依据触发条件的情况，可以控制病毒的感染或破坏动作的频率，使病毒在隐蔽的情况下，进行感染或破坏动作。病毒的触发条件有多种形式，例如日期、时间、发现特定程序、感染的次数、特定中断的调用次数。

主控模块在总体上控制病毒程序的运行，其基本动作如下：①调用感染模块，进行感染；②调用触发模块，接收其返回值；③如果返回真值，执行破坏模块；④如果返回假值，执行后续程序。

感染了病毒的程序运行时，首先运行的是病毒的主控模块。实际上病毒的主控模块除上述基本动作外，一般还做下述工作：①调查运行的环境；②常驻内存的病毒要做包括请求内存区、传送病毒代码、修改中断矢量表等动作，这些动作都是由主控模块进行的；③病毒在遇到意外情况时，必须能流畅运行，不应死锁。下面用伪代码对病毒的结构进行详细描述。

```
program  virus: =
 {1234567:
  subroutine  infect - executable: =
  {loop:file = get - random - executable - file;
  if first - line - of - file = 1234567 then goto loop;
  prepend virus to file;
   }
 subroutine  do - damage: =
{whatever damage is to be done}
subroutine  trigger - pulled: =
{return  true  if  some  condition  holds}
main - program: =
{infect - executable;
 if  trigger - pulled  then  do - damage;
 goto next;}
next:}
```

病毒从主程序开始，先执行 infect-executable 子程序，病毒程序(V)搜索一个未被病毒感染的可执行程序(E)。根据程序开始行有无"1234567"判定程序是否被病毒感染。如果开始行为 1234567，则表示程序已被病毒感染，不再进行传染；如果开始行不是 1234567，则表示程序没有被病毒感染，把病毒(V)放到可执行程序(E)的前面，使之成为感染的文件(I)，PREPEND 语句的作用就是将(V)放到(E)的前面。

接着，病毒程序(V)检查激发条件是否为真，如果为真，则执行 DO-DAMAGE 子程序，即进行破坏，最后(V)执行它所附着的程序；如果激发条件不满足，则执行 NEXT 其他的子程序。当用户要运行可执行程序(E)时，实际上是(I)被运行，它传染其他的文件，然后再像(E)一样运行，当(I)的激发条件得到满足时，就去执行破坏活动，否则除了传染其他的文件要占用一定的系统开销外，(I)和(E)都具有相同的功能。

一个病毒程序的作用，其关键在于动态执行过程中具有病毒传递性。需要指出，病毒并不一定要把自身附加到其他程序前面，也不一定每次运行只感染一个程序。如果修改病毒程序(V)，指定激发的日期和时间，并控制感染的多次进行，则有可能造成病毒扩散到整个计算机系统，从而使系统处于瘫痪状态。

10.1.3 计算机病毒的检测

计算机病毒的检测方法主要有长度检测法、病毒签名检测法、特征代码检测法、校验和法、行为监测法等。这些方法依据的原理不同，实现时所需开销不同，检测范围不同，每种方法均有其自身的优缺点。

1. 长度检测法

病毒最基本的特征是感染性，感染后通常会引起宿主程序的增长，一般增长几百字节。长度检测法，就是从文件长度的非法增长发现病毒。不同病毒使文件增长的长度一般不同，因而从文件增长的字节数可以大致断定文件感染了何种病毒。以文件长度是否增长作为检测病毒的依据，在许多场合是有效的。但是，长度检测法有其局限性，例如某些病毒感染文件时，宿主文件长度可保持不变，因而只检查可疑程序的长度是不充分的。

2. 病毒签名检测法

病毒签名（病毒感染标记）是宿主程序被病毒感染的标记，不同病毒感染宿主程序时，在宿主程序的不同位置放入特殊的感染标记。这些标记是一些数字串或字符串，例如1357、1234、MsDOS、FLU等。不同病毒的病毒签名内容不同、位置不同。经过剖析病毒样本，掌握了病毒签名的内容和位置之后，可以在可疑程序的特定位置搜索病毒签名。如果找到了病毒签名，那么可以断定可疑程序中有病毒、是何种病毒。这种方法称为病毒签名检测方法。

3. 特征代码检测法

病毒签名是一个特殊的识别标记，它不是可执行代码，并非所有病毒都具备病毒签名。病毒本身一般以二进制代码的形式存在，其中存在某一个代码序列用于唯一标识一个病毒，称为特征代码。某些病毒判断宿主程序是否受到感染是以宿主程序中是否含有特征代码作为判断依据，因此，反病毒专家也采用了类似的方法检测病毒，在可疑程序中搜索某些特殊代码，称为特征代码检测法。特征代码检测法被普遍用于各商业反病毒工具软件中，是检测已知病毒的最简单、开销最小的方法。实施特征代码需要经过以下两个步骤。

1）建立特征代码库

专业反病毒人员首先采集病毒样本，通过分析、抽取得到特征代码。抽取的特征代码要具有特殊性，能够在大范围的匹配中唯一标识病毒，因此特征代码的长度不应太短，但为了提高匹配效率、减少数据存储负担，特征代码的长度也不应太长，一般为十几字节。特征代码被加入特征代码库，库中存储了大量已知恶意代码的特征代码。

一般计算机病毒的特征代码为十几字节，但有的代码由于较为特殊而更短，表10.1给出了几个计算机病毒的特征代码。

表10.1 计算机病毒的特征代码

病毒名称	病毒的特征代码（十六进制）
DISK Killer	C3 10 E2 F2 C5 06 F3 01 FF 90 EB 55
CIH	55 8D 44 24 F8 33 DB 64
ItaVir	48 EB D8 1C D3 95 13 93 1B D3 97
Vecomm	0A 95 4C B3 93 47 E1 60 B4

2）特征代码匹配

根据特征代码库，检测工具对检测目标实施代码扫描，逐一检查特征代码库中的特征代码是否存在。为了加快匹配，特征代码一般也记录了特征代码出现的位置。

特征代码检测法检测较为准确，有利于清除工作，但是，由于该方法不能检测出未知的恶意代码，特征代码库要经常更新，而且在特征代码库尺寸比较大时，匹配开销比较大。

4. 校验和法

很多检测工具都为用户提供了一种文件完整性保护方法——校验和法，它计算文件的校验和，将该校验和写入被保护文件、其他文件或内存中。计算校验和类似于计算散列值。在文件使用过程中，定期地或每次使用文件前，检查根据文件当前内容算出的校验和与原来保存的校验和是否一致，从而发现文件是否感染，这种方法叫校验和法。

这种方法既能发现已知病毒，也能发现未知病毒，但是，它不能识别病毒种类。由于病毒感染并非文件内容改变的唯一的原因，文件内容的改变有可能是正常程序引起的，所以校验和法常常误报警。而且此种方法也会影响文件的运行速度。

5. 行为监测法

病毒在执行过程中可能存在一些特殊的行为，行为监测法就是通过发现它们进行报警。例如一般用户程序很少修改可执行文件，而可执行文件是病毒主要的感染对象，因此一旦有程序要修改可执行文件，可以立即分析这个程序的来历，判断其是否为恶意代码；一些文件型病毒在执行完病毒代码后转而执行原宿主程序，因此存在较大的上下文环境变化，这是病毒的行为特征之一。

采用行为监测法可以识别病毒的名称或种类，也可以检测未知病毒，但是也存在一定的误报。

6. 软件模拟法

一些多态性病毒在每次感染中都变化其病毒代码（例如每次用不同的密钥加密病毒代码），对付这种病毒，特征代码法失效。虽然行为检测法可以检测多态性病毒，但是在检测出病毒后，因为不知病毒的种类，难以做消毒处理。

软件模拟法用可控的软件模拟器模拟恶意代码的执行，在执行中确认恶意代码的特征。该类工具开始运行时，使用特征代码法检测病毒，如果发现隐蔽病毒或多态性病毒嫌疑时，启动软件模拟模块，监视病毒的运行，待病毒自身的密码译码以后，再运用特征代码法来识别病毒的种类。

软件模拟法在执行中代价相对较高，一般仅面向常用方法失效的情况。

7. 感染实验法

感染实验是一种简单实用的检测病毒方法。这种方法的原理是利用病毒的最重要的基本特征——感染特性。计算机病毒在内存驻留期间往往感染那些获得执行或打开的文件。感染实验法的原理是将一些已确定是干净的文件复制到可能含有病毒的系统中，反复执行或打开它们，根据它们长度或内容上的变化来确定病毒的存在。

感染实验法简单、实用，可以检测未知计算机病毒的存在，但一般较难识别病毒的名称。

10.1.4 病毒防御

1. 反病毒软件

反病毒软件是现在最为普遍的病毒防御安全机制。反病毒软件检测恶意代码的方法是利用病毒的特征码，防病毒软件商收集病毒样本并采集其特征码，通常反病毒软件数据库中收录成千上万个病毒特征码，当反病毒软件扫描文件时，将当前文件和病毒特征相比较，检测是否有文件片段和特征码相吻合，以此判断文件是否染毒。

对于基于病毒特征码的检测方法，最大的挑战是反病毒软件只有包含了这个特征码，才能够在系统中发现病毒。这意味着反病毒软件供应商需要收集完备的病毒样本，并且尽快开发出标志它们的特征码分发给用户，而用户则要每隔一段时间下载最新的病毒特征库，即使快速频繁地更新，匹配病毒特征码的方法仍然有不可克服的缺点，因为反病毒软件总是走在病毒的后面，只有等病毒开始传播了，才能够及时收集到样本，而往往病毒已经造成了破坏。

2. 强化配置、加强管理

强化配置的目的是使环境尽可能地不被病毒感染，同时阻止被感染后病毒的传播，例如，可以通过在微软 Word 中进行设置禁止宏的执行，也可以在浏览器中设置禁止下载不可信的插件。一定的管理制度也有利于更好地抵御病毒的侵害，例如加强对光盘、移动硬盘等介质和网络下载的管理可以减少文件病毒的传播。

10.2 特洛伊木马

特洛伊木马(Trojan Horse)简称木马，名称来源于希腊神话《木马屠城记》，在古希腊传说中，特洛伊王子帕里斯访问希腊，诱走了王后海伦，希腊人因此远征特洛伊，由于特洛伊城池牢固易守难攻，希腊军队和特洛伊勇士们对峙长达 10 年之久，希腊将领奥德修斯献了一计，让希腊军队假装撤退，留下一具巨大的中空木马，特洛伊守军不知是计，把木马运进城中作为战利品。夜深人静之际，木马腹中躲藏的希腊士兵打开城门，希腊将士一拥而入攻下了城池，特洛伊沦陷。后人常用“特洛伊木马”这一典故，用来比喻在敌方营垒里埋下伏兵里应外合的活动。

在计算机领域，我们将表面上看似正常的程序，但实际上却隐含着恶意意图的恶意代码称为木马。为什么要把这样的黑客工具叫做特洛伊木马呢？我们可以进行这样的对比：入侵者即黑客，相当于希腊军队，他想要入侵的是某个主机，相当于特洛伊城，而该主机比较安全，黑客无法从正面攻入该主机。因此，黑客使用一个迷惑性的外壳包装了恶意的木马程序，并诱使主机的用户自己将木马程序下载到主机中并执行。当木马执行后，黑客就可以利用木马所突破的通道进入主机中。这是不是非常类似于“特洛伊战争”？这就是之所以称其为特洛伊木马的原因。

10.2.1 木马的功能与特点

特洛伊木马是黑客常用的一种攻击工具，它伪装成合法程序，植入目标系统，对计算机

网络安全构成严重威胁。特洛伊木马程序与一般的病毒不同，它不会“刻意”地去感染其他文件，也就是没有传染性，另外病毒的主要目的是破坏数据，而木马的主要目的是偷窃数据。木马的基本功能如下。

1. 窃取数据

以窃取用户的网游账号、网银账号和密码等重要信息为目的，这是木马最常见的功能。木马可以侦测到一切以明文的形式或缓存在 Cache 中的密码。

2. 远程控制

目前常见的远程控制操作有以下两种：一是实时截取用户屏幕图像，监视远程用户当前正在进行的操作；二是远程桌面控制，通过远程控制窗口或命令，直接控制目标计算机进行相应操作，例如在目标计算机上执行程序、攻击其他计算机等。

3. 远程文件管理

攻击者可通过远程控制对被控主机上的文件进行复制、删除、上传、下载等一系列操作，基本涵盖了 Windows 平台上所有的文件操作功能。

4. 打开未授权的服务

木马程序被植入远程用户后，可以偷偷安装及打开某些服务，使其为攻击者服务。例如打开文件共享服务，这样攻击者就可以下载自己需要的文件；或者打开 FTP 服务，把被控主机设定为 FTP 文件服务器，使其提供 FTP 文件传输服务。

主流木马的功能一般比较强大，诸如 Back Orifice、Sub7、冰河等木马的功能便十分全面。它们不仅可以搜集和窃取用户账户密码等数据信息，还能够用作 Telnet 服务器、HTTP 服务器、远程控制器、键盘记录器等。尤其是一些恶性木马还具有反侦测能力，隐蔽性强。这类木马由于实现功能较多，导致体积较大，一般可以达到 100～300KB。而有些木马被设计用来完成特定操作，为后续入侵提供便利，诸如 ProtectedStorage 木马、Keylogger 木马等。此类木马往往用于窃取初始远程控制能力，为在用户系统中安装功能全面的大型木马提供便利，这些木马功能便比较单一，体积也较小，通常在 10KB 左右。

随着木马技术的发展，出现了形态各异的木马，它们使用不同程序语言编写，运行在不同的平台环境下，采用的技术也不尽相同，但是它们仍然有着许多共同的特点。

(1) 隐蔽性。隐蔽性是木马的突出特点。木马必须采用各种技术隐藏自己，实现长期潜伏于目标机器中而不被用户发现。例如木马通常会设置自身文件的属性为“隐藏”和“系统”，并把文件名改成与系统文件类似来隐蔽自己，或者伪装成图片、文本等非可执行文件。木马通过各种技术实现木马文件隐藏、启动隐藏、进程隐藏、内核模块隐藏和通信隐藏。

(2) 自启动性。自启动性也是木马的重要特征。木马只有伴随系统启动才能更好地完成自身的功能。典型的木马自动加载技术有修改系统“启动”项；修改注册表的相关键值；修改“组策略”；修改系统配置文件(Win.ini、System.ini 和 Autorun.bat 等)；利用文件关联技术和文件劫持技术等，随着木马技术的发展，各种更加先进的自启动技术也不断涌现。

(3) 自动恢复性。许多木马程序不再由单一文件组成，而是由多个木马文件协同工作，具有多重备份。这些文件同时运行，互相检查其他文件是否被删除。一旦发现其他文件被删除，就会将其自动恢复。同时，木马可以使用多个守护进程，这些进程相互监视，达到防止木马进程被关闭的目的。

(4) 易植入性。向目标主机成功植入木马,是木马成功运行、发挥作用的前提。易植入性就成为木马有效性的先决条件。通过电子邮件传播是木马最常见的植入手段。木马程序往往伪装成电子贺卡或图片,欺骗用户打开。与免费共享文件绑定也是木马植入的重要手段。用户下载后,只要运行这些程序,木马就会自动安装。另外,木马也经常利用系统的一些漏洞进行植入。

【例 10-2】 PKZip 木马。

PKZip 是一个被广泛使用的文件压缩程序,在 PKZip 的版本达到 2.04 时,网上出现了 PKZip 300,它看起来像 PKZip 的更新版本(木马具有伪装性),但当用户下载并运行后,PKZip 300 对硬盘立即实施攻击,造成系统破坏。

10.2.2 木马工作机理分析

木马程序大多采用 C/S 架构,其中,服务器端程序潜入目标机内部,获取系统操作权限,接收控制指令,并根据指令或配置发送数据给控制端。服务端程序通常隐藏在一些合法程序或数据当中,例如,游戏软件、工具软件、电子邮件的附件、网页等,某个系统“中了木马”,就是指安装了木马的服务端程序。客户端程序又称为控制端程序,用来远程控制被植入木马的机器,安装在控制者的计算机,它的作用是连接木马服务器端程序,监视或控制远程计算机。

典型的木马工作原理是:当服务器端在目标计算机上被执行后,木马打开一个默认的端口进行监听,当客户机向服务器端提出连接请求时,服务器上的相应程序就会自动运行来应答客户机的请求,服务器端程序与客户端建立连接后,由客户端发出指令,服务器在计算机中执行这些指令,并将数据传送到客户端,以达到控制主机的目的。这种由控制端向服务器端发起通信的木马称为正向连接型木马,是当前木马最广泛采用的方式,但是由于防火墙对于由外部向内部发起的连接过滤严格,这种通信方式不能穿透防火墙。为了规避防火墙的限制,又出现了反向连接型木马,这种木马通信时由服务器端向控制端发起连接。木马服务器端程序首先通过一个指定的第三方网址获取木马攻击者的 IP 地址,然后根据该 IP 地址向控制端发起连接,例如灰鸽子木马就采用这种连接方式。

木马的一般攻击过程包括配置木马、传播木马、运行木马、信息泄露、建立连接、远程控制 6 个步骤实现。

1. 配置木马

一般来说,一个设计成熟的木马都有木马配置程序,从具体的配置内容来看,主要是为了实现以下两个方面的功能。

(1) 木马伪装:木马配置程序为了在服务端尽可能最好地隐藏木马,可以通过配置采用多种伪装手段,如修改图标、绑定文件、定制端口和自我销毁等。

(2) 信息反馈:木马配置程序可以配置信息反馈方式或地址,如设置信息反馈的邮件地址、IRC 号和 ICQ 号等。

2. 传播木马

木马传播主要有两种方式:一种是通过 E-mail,控制端木马程序以附件的形式夹在邮件中发送出去,收信人只要打开附件系统就会感染木马;另一种是软件下载,一些非正规的

网站以提供软件下载为名义，将木马绑定在软件安装程序上，下载后，只要一运行这些程序，木马就会自动安装。

3. 运行木马

服务端用户运行木马或绑定木马的程序后，木马就会自动进行安装。如首先将自身拷贝到 Windows 的系统文件夹中 C:\WINDOWS 或 C:\WINDOWS\SYSTEM 目录下，然后在注册表、启动组、非启动组中设置好木马的触发条件，这样木马的安装就完成了，安装后就可以启动木马。一般木马会随系统自动启动。

4. 信息泄露

一般来说，设计成熟的木马都有一个信息反馈机制，所谓信息反馈机制是指木马成功安装后会收集一些服务端的软硬件信息，并通过 E-mail、ICQ 的方式告知控制端用户。

5. 建立连接

在服务器端已经安装了木马程序，在线并启动木马的前提下，控制端可以通过木马端口与服务器建立连接。

6. 远程控制

木马连接建立后，控制端和服务器端会出现一条通道，控制端上的控制程序可通过这条通道与服务器上的木马程序取得联系，并通过木马程序对服务器端进行远程控制，一般可以完成窃取密码、修改注册表和系统操作等功能。

10.2.3　木马实例-冰河木马

1999 年，西安电子科技大学学生黄鑫开发了冰河木马。在设计之初，开发者的本意是编写一个功能强大的远程控制软件。但一经推出，就依靠其强大的功能成为黑客们发动入侵的工具，曾经创造了黑客使用量最大、计算机感染数量最多的奇迹，并结束了国外木马一统天下的局面，成为国产木马的标志和代名词。

冰河属于远程控制型木马，它使用了客户端/服务器的方式进行工作，其文件组成主要包括客户端程序 G_Client.exe 与服务器程序 G_Server.exe，如图 10.1 所示。冰河的服务器程序用于注入到目标计算机，客户端程序则用来设置服务器程序和监控目标计算机。



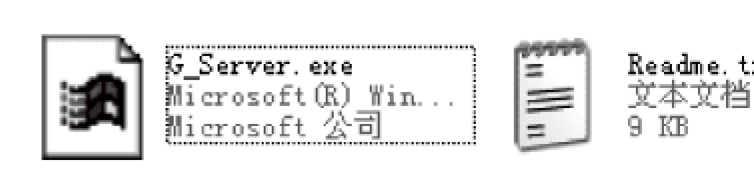


图 10.1　冰河的文件组成

1. 配置木马

如果要利用冰河进行网络入侵，首先需要使用冰河客户端对服务器程序 G_Server.exe 进行配置。选择客户端程序中【设置】主菜单中的【配置服务器程序】子菜单，弹出【服务器配置】对话框（如图 10.2 所示），可以看到冰河的木马配置具体包括【基本设置】、【自我保护】和【邮件通知】三个部分。

图 10.2 中当前打开的选项卡是【基本设置】，界面左下方的【待配置文件】指明了所配置

图 10.2　服务器配置界面

的对象是服务器端程序 G_Server.exe。黑客也可以事先对服务器端程序进行重命名，再使用客户端进行配置操作，有助于增强隐蔽性。

【安装路径】选项用于确定木马服务器端程序植入目标计算机后的安装位置，默认是安装在 C:\Windows\system 系统目录下。同时，黑客可以设置【文件名称】和【进程名称】。文件名称指的是木马服务器端程序在感染主机上的名称，默认为 KERNEL32.EXE。进程名称是木马服务器端程序在感染主机上运行时在进程栏中显示的名称，默认为 Windows。两者都具有很强的欺骗性，一般的计算机用户如果通过手工查看的方式查找木马，很容易误认为木马是正常的系统文件，而不会及时进行清除。

设置【访问口令】主要是考虑到有很多黑客利用冰河进行网络入侵，每个黑客都不希望自己侵入的主机被其他黑客利用。通过设置访问口令，试图访问木马的用户必须输入正确的口令才能够对木马实施远程控制，避免了攻击成果被其他黑客所利用。

设置【敏感字符】可以指定冰河在植入用户计算机后收集并保存与敏感字符相关的信息。常见的敏感字符包括“口令”、“密码”、“登录”、“账户”等，黑客可以根据自己的需求进行设置。

【提示信息】一项可以指定冰河在受害者主机上运行时弹出的对话框信息，该设置项默认为空，即木马程序运行时没有任何提示。随着计算机用户安全意识的不断增强，用户如果运行了一个文件，但文件没有任何响应，很可能会怀疑文件是木马等恶意程序，进而利用防病毒软件查杀。通过设置一些欺骗性的提示信息，如“文件校验错误，请重新下载”等，反而不容易引起用户怀疑，为木马的植入和运行提供便利。

【监听端口】的设置是配置木马阶段的核心工作。监听端口指明了木马服务器端程序进行监听时所使用的端口。黑客需要根据监听端口找到木马的服务器端程序实施远程控制。冰河服务器端程序默认的监听端口是 7626，黑客可以根据需要对监听端口进行灵活设置。

冰河的基本设置还要求黑客配置【是否自动删除安装文件】。如果该项是勾选的，在默认情况下，用户运行 G_Server.exe 文件以后，在 C:\Windows\system 系统目录下将生成文件 KERNEL32.EXE。程序 KERNEL32.EXE 作为木马服务器端程序进行破坏活动，同时，G_Server.exe 作为过渡性的安装文件将被自动删除。

【禁止自动拨号】一项默认是勾选的。如果允许自动拨号，冰河的服务器端程序将在每次开机时自动拨号上网将收集到的信息发送给黑客。由于这种行为过于明显，计算机用户很容易发现系统异常。从隐蔽性的角度考虑，黑客一般会禁止木马的自动拨号功能。

冰河木马配置的第二部分是【自我保护】选项卡，其设置界面如图 10.3 所示。该部分的第一项设置是【写入注册表启动项】，如果勾选该项，冰河的服务器端程序会通过在注册表的启动项中增加冰河服务器程序的信息，保证冰河能够在感染主机开机时自动加载运行。冰河木马有多个版本，注册表的修改涉及的启动项主要是 HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN 和 HEKY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENT VERSION\RUNSERVICE 两项。

图 10.3　冰河木马的自我保护配置界面

黑客可以设置【是否将冰河与文件类型相关联，以便被删除后在打开相关文件时自动恢复】选项。该选项能够提高冰河服务器端程序在感染主机上的存活能力。

黑客可以对冰河的服务器端程序进行设置，将冰河程序 SYSEXPLR.EXE 与指定类型的文件关联在一起。例如，如果所关联的是 txt 类型的文件，txtfile 子键下 shell\open\command 键的默认值将被修改为“C:/windows/system/SYSEXPLR.EXE %1”。文件关联建立以后，每当用户打开 txt 文件，冰河程序 SYSEXPLR.EXE 将被激活，该程序会判断系统中的 KERNEL32.EXE 是否还存在。如果 KERNEL32.EXE 运行正常，SYSEXPLR.EXE 将调用系统中的 NOTEPAD.EXE 打开 txt 文件，同时自动退出，用户感觉不到任何异常。如果被激活的 SYSEXPLR.EXE 在系统中找不到 KERNEL32.EXE，它将重新生成 KERNEL32.EXE。这使得从感染主机上彻底清除冰河变得非常困难。

冰河木马配置的第三部分是【邮件通知】选项卡，该部分的设置界面如图 10.4 所示。邮件通知可以让黑客及时了解自己散播到网络上的冰河感染对象的具体情况。【SMTP 服务器】一项由黑客填充，设置为冰河用来发送邮件的 SMTP 服务器，【接收邮箱】为黑客接收信息的邮箱。黑客根据自己的需要，可以选择【系统信息】、【开机口令】、【缓存口令】、【共享资源信息】中的一项或者多项作为邮件内容，由冰河服务器端程序通过邮件发送给自己。

服务器配置
基本设置 自我保护 邮件通知
如果不需要发送动态IP，请将编辑框清空
SMTP服务器： 接收信箱：
邮件内容
系统信息 开机口令 缓存口令 共享资源信息
待配置文件： G_SERVER.EXE ... 确 定 关 闭

图 10.4 冰河木马的邮件通知设置界面

2. 传播木马

在黑客根据自己的需求配置好木马的服务器端程序以后,下一步的工作就是将所配置的木马程序传播出来,让尽可能多的计算机用户感染木马。传播木马的方式多种多样,例如网站挂马、利用电子邮件传播、利用聊天软件传播和在用户下载中传播等。

3. 运行木马

冰河木马的服务器程序在植入计算机后,会根据黑客在配置木马阶段进行的设置适时运行,为黑客实施远程控制提供服务。根据配置木马相关知识可知,冰河服务器程序根据配置可以开机自启动,可以与文件类型相关联,以便被删除后再打开相关文件时自动恢复。

4. 信息反馈

木马获得运行机会以后,需要把感染主机的一些信息反馈给配置和散播木马的黑客。从远程控制的角度看,黑客最关注的信息是受害主机的 IP 地址。因为黑客只有掌握感染主机的 IP 地址,才能与主机上运行的木马程序建立连接,实施远程控制。

5. 建立连接

在完成信息反馈的操作之后,下一步就是建立连接。根据信息反馈方式的不同,木马的服务器端程序与木马的客户端程序建立连接的方式可以划分为两种。第一种是木马的服务器端程序进行监听,木马的客户端程序发起连接请求。采用这种方式,需要木马的服务器端程序将感染主机的信息反馈给黑客,由黑客通过木马的客户端程序建立连接。第二种的连接方式是采用反向连接技术,由木马的客户端程序进行监听,木马的服务器端程序发起连接请求。这种连接方式需要木马的服务器端程序掌握木马客户端主机的地址信息,以保证网络连接能够成功建立。

冰河木马采用的是第一种方式,除了可以通过邮件通知的方式将感染主机的信息反馈给黑客外,黑客可以使用扫描工具主动在网络上查找感染主机。黑客在木马配置阶段可以指定冰河服务器程序在哪一个端口进行监听,为了避免和感染主机上的正常程序冲突,一般选择比较特殊的监听端口,如冰河的默认端口是 7626。如果黑客在配置木马时指定木马的服务器端程序在 9999 端口进行监听,在木马程序传播出去以后黑客可以在网络上使用端口

扫描工具，查找开放 9999 端口的主机。因为主机正常情况下一般不会使用 9999 端口进行监听，开放该端口的主机很可能感染了木马程序，黑客可以尝试利用木马客户端程序对主机进行连接和控制。这种利用网络扫描的方法，其优点是不依赖于信息反馈，对于采用动态 IP 地址的感染主机也能够有效控制。其缺点也很明显，网络扫描具有明显的攻击特征，很容易被入侵检测系统等安全防护设备发现并受到限制。冰河客户端程序中提供了扫描感染主机的工具，单击工具栏中的自动搜索，弹出【搜索计算机】的界面（如图 10.5 所示），设置好搜索范围后单击【开始搜索】，可以搜索到哪些主机感染了冰河木马。

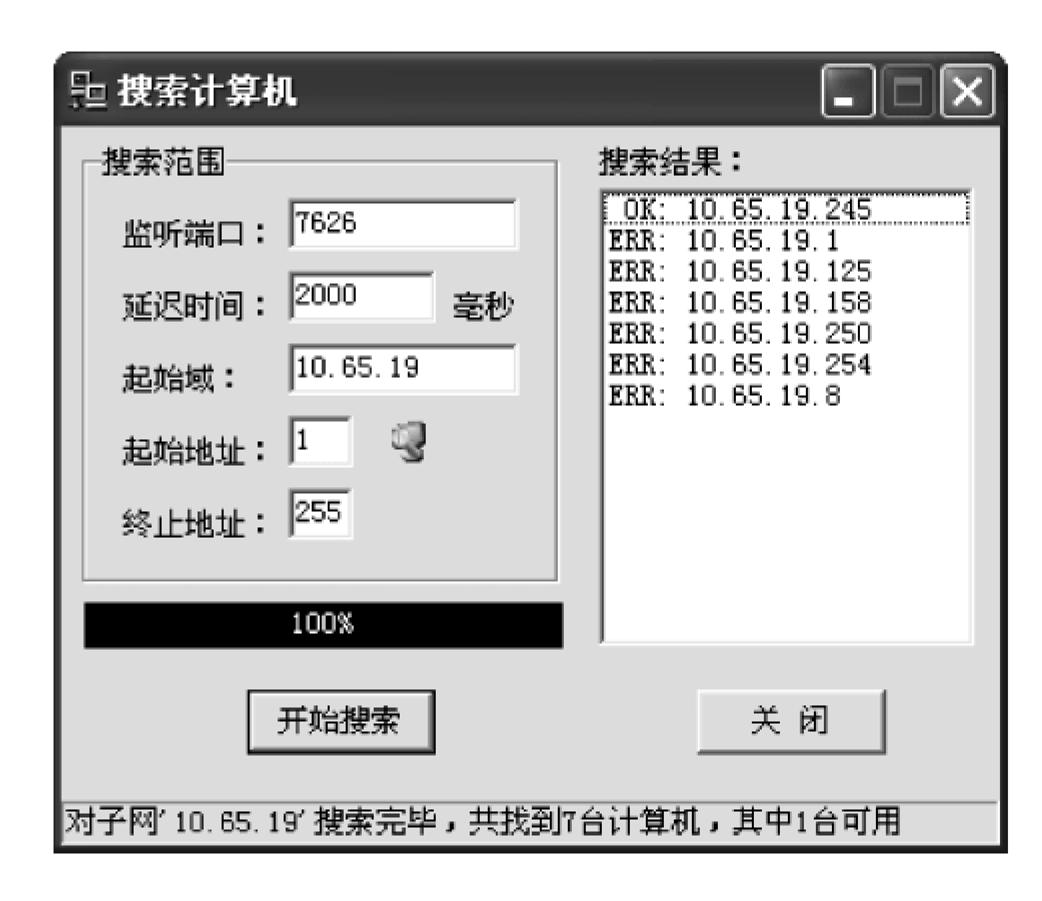


图 10.5　搜索感染计算机的界面

搜索到感染主机后，单击工具栏中的【添加主机】按钮，输入计算机 IP 地址、监听端口和访问口令，单击确定后，冰河客户端就开始尝试与该主机中的服务器程序建立连接，如果成功建立连接，界面右边会显示该主机内的磁盘盘符（如图 10.6 所示）。



图 10.6　添加感染主机并建立连接

6. 远程控制

木马的客户端程序与木马的服务器程序一旦建立连接，黑客实际上就获得了感染主机的控制权。冰河木马可以实现以下一些远程控制操作：

（1）自动跟踪目标主机屏幕变化，同时可以完全模拟键盘及鼠标输入，即在同步感染主机屏幕变化的同时，监控端的一切键盘及鼠标操作将反映在感染主机的屏幕上（局域网适

用)；单击冰河客户端工具栏中的【控制屏幕】按钮，设置好图像参数后，在弹出的窗口中可以查看和控制目标主机的屏幕。

(2) 记录各种口令信息：包括开机口令、屏保口令、各种共享资源口令及绝大多数在对话框中出现过的口令信息。获取系统信息：包括计算机名、注册公司、当前用户、系统路径、操作系统版本、当前显示分辨率、物理及逻辑磁盘信息等多项系统数据。启动键盘记录等。

(3) 限制系统功能：包括远程关机、远程重启计算机、锁定鼠标、锁定系统热键及锁定注册表等多项功能限制。

(4) 远程文件操作：包括创建、上传、下载、复制、删除文件或目录，文件压缩，快速浏览文本文件，远程打开文件(提供了 4 种不同的打开方式——正常方式、最大化、最小化和隐藏方式)等多项文件操作功能。

(5) 注册表操作：包括对主键的浏览、增删、复制、重命名和对键值的读写等所有注册表操作功能。

(6) 发送信息：以 4 种常用图标向感染主机发送简短信息。

(7) 点对点通信：以聊天室形式同感染主机进行在线交谈。

在冰河客户端界面中选择【命令控制台】选项卡，在左侧的命令树中可以选择相应的控制命令实现远程控制操作(如图 10.7 所示)。



图 10.7 冰河命令控制台界面

10.2.4 木马的检测与防范技术

木马的查杀，最简单的方法是利用杀毒软件，目前大多数杀毒软件，例如卡巴斯基、诺顿、小红伞、Avast 等都能有效删除大多数木马。另外也可以使用专门的木马专杀工具来杀除，例如木马克星、Spy Sweeper、360 安全卫士等软件。由于杀毒软件和专杀工具的更新速度通常滞后于新木马的出现，因此有必要掌握木马的检测和手动清除方法。

1. 检查本地文件

在我的电脑窗口下打开【工具】|【文件夹选项】|【查看】|【显示所有文件和文件夹】，可以看到隐藏的文件，再选择【显示已知文件类型扩展名】，查看是否存在多扩展名的程序，如果有就可能是木马文件，然后检查系统文件是否都处于正常的系统文件夹内，如果存在多个同样的系统文件，那么就要注意查看是否为木马文件。

2. 检查端口及连接

由于木马一般需要通过端口进行通信（如冰河木马使用 7626 端口进行监听），因此检查系统开放端口及连接是辅助判断木马的一个重要依据。用户可以通过系统自带的 netstat 命令查看系统的开放端口和 TCP/UDP 连接，辅助使用 Fport 等工具具体查看进程与端口的映射关系，来判断系统是否有木马存在。

3. 检查系统进程

用户可以利用 Windows 系统自带的任务管理器检查系统的活动进程，观察有没有陌生的进程，重点注意一些 CPU 占用率较高的进程。另外借助一些专门工具如 ProcessExplorer，可以检查更加详尽的进程列表信息，以此来判断进程的合法性。找到对应木马文件后，就可以把木马进程结束。

4. 检查注册表

重点检查注册表启动项和系统服务相关键值。运行 regedit 命令打开注册表，展开 HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\和 HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\下的所有以 Run 开头的项，检查其下是否有新增的和可疑的键值。找到木马程序的文件名后搜索整个注册表，找出所有对应的项，然后删除或修改注册表里的相应内容，并将安装路径所指示的文件删除。

5. 检查系统配置文件

（1）检查 Win. ini。在 C:\WINDOWS 目录下有一个配置文件 Win. ini，在它的 windows 字段中有启动命令“load=”和“run=”，在一般情况下，“=”后面是空白的，如果有启动程序，很可能就是木马。

（2）检查 System. ini。在 System. ini 中，其[boot]字段的“shell=explorer. exe”是加载木马的常见位置。如果该字段变为：“shell=explorer. exe 某一程序名”，那么后面跟着的那个程序就是木马。另外，System. ini 的[386Enh]字段内的“driver=路径\程序名”也是木马常更改的项。

防范木马攻击最基本的方法就是安装杀毒软件和防火墙，防火墙可以主动拦截各种应用程序的网络连接，杀毒软件可以查杀绝大部分的木马。用户可以辅助使用专业的木马防护工具如 360 安全卫士等实现对系统和网络的监控。对于一般用户来说，可以采取的主要措施是完善系统安全和提高用户安全意识，具体包括：

（1）安装杀毒软件和个人防火墙，并及时升级。

（2）为个人防火墙设置好安全等级，防止未知程序向外传送数据。

（3）使用安全性比较好的浏览器和电子邮件客户端工具。

（4）如果使用 IE 浏览器，应该安装卡卡安全助手或 360 安全浏览器，防止恶意网站在

自己电脑上安装不明软件和浏览器插件,以免被木马趁机侵入。

(5) 不要随意下载和运行来历不明的软件,安装软件之前先用杀毒软件查杀。

(6) 不随意散播个人电子邮箱地址,不要任意执行电子邮件中的附件。

(7) 不登录陌生网站,禁用浏览器的"ActiveX 控件和插件"以及"Java 脚本"功能,禁用文件系统对象 FileSystemObject,防止恶意站点网页木马全自动入侵。

10.3 蠕　　虫

蠕虫这个生物学名词在 1982 年由施乐帕克研究中心(Xerox PARC)的 John F. Shoch 等人最早引入计算机领域。在计算机领域,蠕虫是一种可以自我复制的代码,并且通过网络传播。蠕虫袭击一台计算机,并在完全控制后,将这台计算机作为宿主,进而扫描并感染其他脆弱的系统,当这些新目标被蠕虫控制后,这种贪婪的行为会继续,蠕虫采用递归的方式进行传播,按照指数增长的方式复制自己,进而感染更多系统。几乎每次蠕虫爆发都会造成巨大的损失,发生过的几件重大事件如下。

(1) 1988 年 11 月,Morris 蠕虫爆发,仅仅在几天内就感染了 6000 台以上的 Internet 服务器,并且使得这些服务器几近瘫痪,严重破坏了当时的互联网;

(2) 2001 年 7 月 19 日,CodeRed 蠕虫爆发,在爆发后仅仅 9h 内就感染了 25 万台计算机,直接经济损失高达 20 亿美元,并且在随后几个月内该蠕虫产生了威力更强的几个变种,其中 CodeRed Ⅱ造成的损失估计超过 12 亿美元;

(3) 2001 年 9 月 18 日,Nimda 蠕虫爆发,也是在仅仅几个小时内,Nimda 造成的损失评估高达 26 亿美元。在 CERT 的声明中,2001 年被称为 Year of the Worm;

(4) 2003 年 1 月 25 日,在亚洲、美洲、欧洲爆发的 slamme 蠕虫在 10min 内感染了 92%的网络服务器。

近年来,蠕虫的活动不仅局限于传统的简单感染存在漏洞的结点,而且它们会把这些结点组织起来形成破坏力更强的僵尸网络。

10.3.1 定义

蠕虫类恶意代码是具有自我复制(Self-Replication)和主动传播(Active-Propagation)能力并在网络上尤其是在 Internet 上进行扩散的恶意代码。网络蠕虫(简称蠕虫)以其多样化的传播途径,使它具有传播速度快、发生频率高、覆盖面广以及造成的危害大等特点。

计算机蠕虫和计算机病毒都具有传染性和复制功能,这两个主要特性是一致的,但是从传染机制和工作方式上看,二者有很大区别,蠕虫是通过网络传播,而病毒不一定要通过网络传播。病毒主要感染的是文件系统,在其传染的过程中,计算机使用者是传染的触发者,需要用户运行一个程序或打开文档以调用恶意代码,而蠕虫主要利用计算机系统漏洞(Vulnerability)进行传染,不需要宿主文件,搜索到网络中存在漏洞的计算机后主动进行攻击,在传染的过程中,无须通过人为干预。表 10.2 列举了病毒和蠕虫的区别。

表 10.2　病毒和蠕虫的区别

	病　毒	蠕　虫
存在形式	寄生	独立存在
复制机制	插入到宿主程序(文件)中	自身的拷贝
传染机制	宿主程序运行	系统存在漏洞
攻击目标	针对本地文件	针对网络上的计算机
触发传染	计算机使用者	程序自身
影响重点	文件系统	网络性能、系统性能
防治措施	从宿主文件中摘除	为系统打补丁

【例 10-3】“红色代码”(RedCode)蠕虫。

2001 年出现的“红色代码”(RedCode)蠕虫利用微软公司的 IIS(Internet Information Server)系统漏洞进行感染,它使 IIS 服务器处理请求数据包时溢出,导致把此数据包当作代码运行,蠕虫驻留后再次通过此漏洞感染其他 IIS 服务器,造成网络带宽性能急剧下降。

10.3.2　蠕虫的结构和工作机制

蠕虫功能模块划分为主体功能模块和辅助功能模块。主体功能模块用来实现蠕虫自我传播,包括信息搜集模块、探测模块、攻击模块和自我推进模块。其中,信息采集模块的作用是对目的主机或网络进行攻击前的信息收集。探测模块完成对特定主机的脆弱性检查,检查结果用于攻击模块决策对目标的攻击方式,如 CodeRed 蠕虫和 Slammer 蠕虫会对随机生成的 IP 地址进行漏洞检测。攻击模块利用获得的安全漏洞,建立传播途径。自我推进模块在不同的感染目标中进行蠕虫的自我复制。

辅助功能模块可以增强蠕虫的破坏力,是对除主体功能模块之外的其他模块的归纳,包括实体隐藏模块、宿主破坏模块、通信模块、远程控制模块和自我更新模块。实体隐藏模块主要实现对蠕虫各部分实体的隐藏,以提高蠕虫的生存能力。宿主破坏模块用来破坏被感染系统或网络的正常运行。通信模块用来实现蠕虫间、蠕虫和黑客间的交流,这是未来蠕虫的发展趋势,也是目前流行的僵尸网络的形成机制。远程控制模块的功能是调整蠕虫行为,控制被感染主机,实现蠕虫编写者的命令。自我更新模块使蠕虫每隔一定的时间自动下载更新代码和传播策略。

蠕虫利用操作系统和应用软件的漏洞进行自我复制和传播。例如,“红色代码”蠕虫利用了微软 IIS 服务器软件的漏洞(idq.dll 缓冲区溢出)进行传播;SQL 蠕虫王病毒利用了微软的数据库系统的一个漏洞进行大肆攻击;Conficker 蠕虫则借助闪存、利用微软的 MS08-067 漏洞进行传播。

尽管这些蠕虫实现的具体细节不同,但是从蠕虫释放到最终在网络中传播的过程基本相同。它们的攻击行为大体分为 4 个阶段:目标信息收集、扫描探测、攻击渗透和自我推进。蠕虫传播的工作机制如图 10.8 所示。信息搜集主要完成对本地和目标结点主机的信息汇集;扫描探测主要完成对具体目标主机服务漏洞的检测,攻击渗透利用已发现的服务漏洞实施攻击;自我推进完成对目标结点的感染。

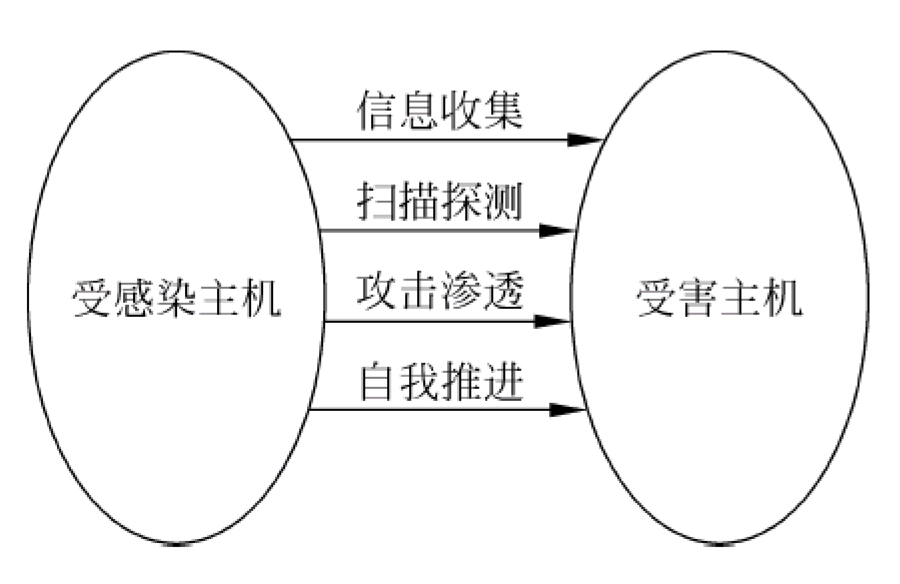


图 10.8　网络蠕虫的工作机制

10.3.3 蠕虫的防范

1. 安装反病毒软件

反病毒软件能够阻止各种形式的恶意代码，也包括蠕虫，用户需要及时更新病毒库提高预防能力。

2. 及时配置补丁程序

由于越来越多的网络蠕虫依靠操作系统自身的漏洞进行传播，因此及时对操作系统的漏洞进行打补丁操作已经成为当前网络蠕虫病毒防治的一个重要环节。在有条件的局域网环境中可以安排专门的更新服务器对局域网内部所有的主机进行集中升级操作。

3. 阻断任意的输出连接

一旦蠕虫侵占了系统，常常通过建立输出连接扫描其他潜在的受害者，进而试图传播。应该严格限制所有来自公共访问系统的输出连接，例如 Web、DNS、E-mail、FTP 等。许多单位严格过滤连入的数据，但是却忘记了输出连接，从而使感染蠕虫者成为蠕虫散布者。

4. 建立事故响应机制

对于一些重要系统来说，建立计算机事故响应小组也是很有必要的，具有明确的处理流程以对抗计算机攻击者、蠕虫和其他恶性事件。

10.4 小　　结

常见的恶意代码包括计算机病毒、特洛伊木马和蠕虫等，它们以各种方式侵入计算机信息系统，它们的攻击威力越来越大、攻击范围也越来越广。本章介绍了以上 3 种恶意代码的工作原理和检测、防范技术。需要注意的是，恶意代码的发展变化很快，一些新的恶意代码类型不断涌现，需要不断研究新的分析检测技术以抵御和防范恶意代码攻击。

习　　题

一、填空题

1. 计算机病毒具有的特征包括________、________、________、________和________。
2. 病毒一般由________、________、________、________和主控模块构成。
3. 计算机病毒按其寄生方式可分为两类：________和________。
4. 木马的功能有________、________、________、________等。
5. 某个系统“中了木马”，就是指安装了木马的________。
6. 蠕虫主要利用计算机系统________进行传染。
7. 蠕虫的攻击行为大体分为 4 个阶段，分别是________、________、________、________。

二、选择题

1. 以下(　　)不是计算机病毒的基本特征。

A. 潜伏性　B. 可触发性　C. 免疫性　D. 传染性

2. 计算机病毒的构成模块不包括(　　)。

A. 感染模块　B. 触发模块　C. 破坏模块　D. 加密模块

3. 计算机病毒常用的触发条件不包括(　　)。

A. 日期　B. 访问磁盘次数　C. 屏幕保护　D. 启动

4. 关于计算机病毒的传播途径,下列说法错误的是(　　)。

A. 通过邮件传播　B. 通过光盘传播

C. 通过网络传播　D. 通过电源传播

5. 如果发现某文件已染上病毒,恰当的处理方法是(　　)。

A. 停止使用,使其慢慢消失　B. 将该文件复制到 U 盘上使用

C. 用消毒液消毒　D. 用反病毒软件清除病毒

6. 以下不属于木马检测方法的是(　　)。

A. 检查端口及连接　B. 检查系统进程

C. 检查注册表　D. 检查文件大小

三、简答题

1. 请简述计算机病毒的工作过程。
2. 如何检测计算机病毒?
3. 日常如何预防计算感染病毒?
4. 木马采用反向连接的原因是什么?
5. 简述木马的攻击过程。
6. 计算机病毒和蠕虫的区别是什么?
7. 简述网络蠕虫的工作机制。
8. 拓展阅读:查询相关资料,了解冰河木马的原理。

第 11 章　应用系统安全

应用系统缺陷是影响计算机安全的重要因素,随着应用系统数量和功能复杂性的不断增加,其潜在的安全隐患也不断增多,应用系统中安全漏洞的报告逐年增长。事实上目前恶意代码的攻击大多是利用了应用软件,尤其是网络应用软件的安全漏洞。

应用系统开发的复杂性是由程序复杂性、需求复杂性以及设计复杂性等各因素构成的,因此在应用系统开发周期的各个阶段都可能发生错误或缺陷,而这些错误或缺陷就直接导致了应用系统安全漏洞的形成。应用系统安全漏洞一旦被发现,攻击者就可利用此漏洞获得计算机系统的控制权限,使其能够在未授权的情况下访问或破坏系统,从而极大地危害到计算机系统的安全。也就是说应用系统安全漏洞可以被恶意攻击者用来突破系统的安全策略或措施,从而访问或破坏未经授权的系统资源和数据。常见应用系统安全漏洞有缓冲区溢出漏洞、格式化字符串漏洞、整数溢出漏洞等。

本章 11.1 节介绍缓冲区溢出漏洞的概念及利用漏洞攻击原理,11.2 节、11.3 节分别简要介绍格式化字符串漏洞和整数溢出漏洞及其危害,11.4 节介绍发掘应用系统安全漏洞的几种方法,随着当前越来越多的应用面向 Web 平台开发、部署、使用,11.5 节讨论 Web 应用安全威胁与防御措施。

11.1　缓冲区溢出

自从缓冲区溢出漏洞被发现以来,缓冲区溢出攻击一直是网络攻击事件中用得最多的一种攻击方式。世界上第一个缓冲区溢出攻击——Morris 蠕虫,曾造成全球 6000 多台网络服务器瘫痪;2003 年 8 月席卷全球的“冲击波”和 2004 年 5 月出现的“震荡波”分别利用了 Windows 系统 RPC DCOM 和 LSASS 服务的缓冲区溢出漏洞进行攻击。目前,利用缓冲区溢出漏洞进行的攻击已经占到了整个网络攻击的一半以上。

11.1.1　缓冲区溢出的概念

缓冲区是程序运行时在内存中临时存放数据的地方。缓冲区就如一个水杯,如果向其中加入太多的水后,水就会溢出到杯外。缓冲区溢出是因为人们向程序中提交的数据超出了数据接收区所能容纳的最大长度,从而使提交的数据超过相应的边界而进入了其他区域。如果是人为蓄意向缓冲区提交超长数据从而破坏程序的堆栈,使程序转而执行其他指令或对系统正常运行造成了不良影响,那么我们就说发生了缓冲区溢出攻击(Buffer Overflow)。缓冲区溢出主要包括基于堆栈的缓冲区溢出、基于堆的缓冲区溢出以及基于数据段的缓冲区溢出。一般来说,堆和数据段的缓冲区溢出很难被攻击者利用,而堆栈上的缓冲区溢出漏洞容易被利用,具有极大的危险性。

C 语言的标准库是多数缓冲区溢出问题的根源所在,尤其是一些对字符串进行操作的

库函数,例如要从标准输入中读取一行输入数据时可运用的 gets()函数,它会一直读入数据直到遇到换行字符或 EOF 字符,并且不会检查缓冲区边界。例如以下这段代码:

```
char buffer[512];
gets(buffer)
```

这段代码中定义了一个长度为 512 的字符数组,接着输入数据被 gets()函数读取并放入 buffer 数组中,所读取的数据一般情况下可能都小于 512 个字符,但如果一些用户例如攻击者在此输入大于 512B 的数据,此时输入数据将超出 buffer 空间而覆盖其他内存数据,从而发生缓冲区溢出。具有类似问题的标准库函数还有 strcat()、sscan()、strcpy()、scan()、fscanf()、sprintf()、vscanf()、vsscanf()、vfscanf()等。

11.1.2　缓冲区溢出攻击原理及防范措施

为了了解缓冲区溢出的机理,先介绍处理器处理机器代码的情况。处理器中有一些特殊的寄存器用来存储程序执行时的信息,主要有以下 3 个。

(1) EIP:扩展指令寄存器,用于存放下一条要执行的指令的地址。

(2) EBP:扩展基址寄存器,存储的是栈底指针,通常称为栈基址。

(3) ESP:扩展堆栈寄存器,用来存放栈顶指针。

计算机内的程序是按如图 11.1 所示的形式存储的。

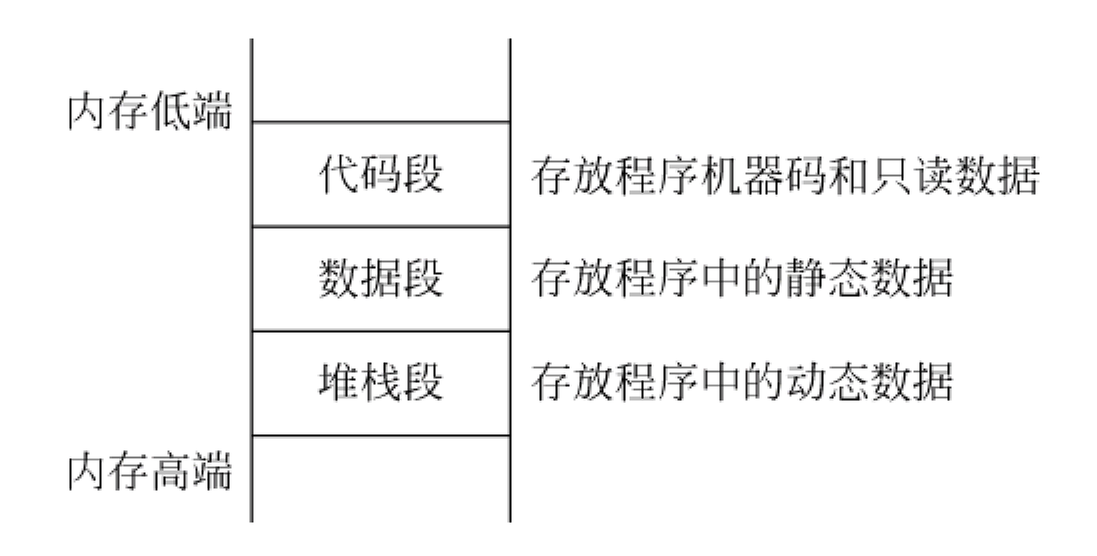


图 11.1　程序在内存中的存储

(1) 代码段:存放程序汇编后的机器代码和只读数据,这个段在内存中一般被标记为只读,任何企图修改这个段中数据的操作将引发一个 Segmentation Violation 错误。

(2) 数据段:数据段中存放的是静态变量。

(3) 堆栈段:在函数调用时存储函数的入口参数(即形参)、返回地址和局部变量等信息。

当一个函数被调用的时候,系统总是先将被调用函数所需的参数以逆序方式入栈,然后将指令寄存器 EIP 中的内容(即返回地址)入栈,再把基址寄存器 EBP 压入堆栈,最后为被调用函数内的局部变量分配所需的存储空间,从而形成如图 11.2 所示的堆栈结构。



图 11.2　栈帧的一般结构

缓冲区溢出攻击通常会带来以下后果：过长的字符串覆盖了相邻的存储单元而造成程序异常，严重的会造成死机、系统或进程重启等；可让攻击者执行恶意代码或特定指令，甚至获得超级权限等，从而引发其他的攻击。

1. 破坏程序正常运行

我们来看一段简单程序的执行对堆栈的操作以及溢出产生的过程。

```
#include<stdio.h>
int main()
{
  char name[16];
  gets(name);
  for(int i=0;i<16 && name[i] ;i++)
    printf(name[i]);
}
```

编译上述代码，输入“hello world!”，结果会输出 hello world!，其中对堆栈的操作是先在栈底压入返回地址，接着将栈指针 EBP 入栈，将当前 ESP 的值写入 EBP，此时 EBP 等于现在的 ESP，之后 ESP 减 16，即向上增长 16B，用来存放 name[]数组，现在堆栈的布局如图 11.3 所示。

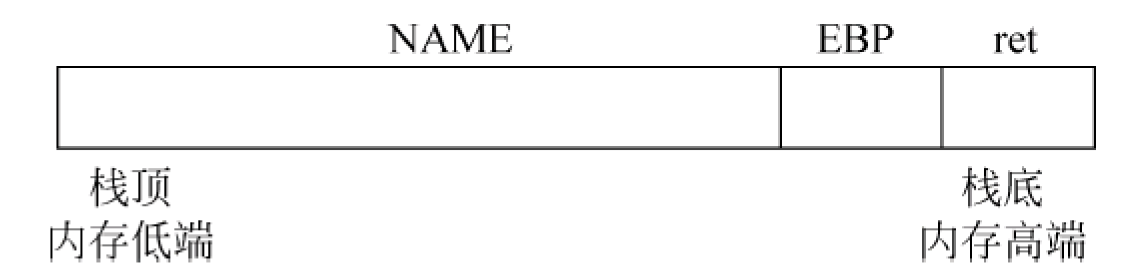


图 11.3 程序运行之初堆栈的状态

执行完 gets(name)之后，堆栈中的内容如图 11.4 所示。

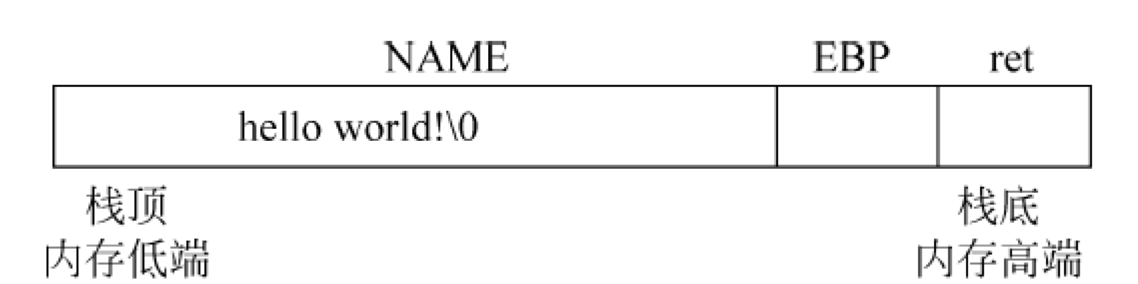


图 11.4 运行完 gets(name)后堆栈的状态

最后，从 main 返回，弹出 ret 里的返回地址并赋值给 EIP，CPU 继续执行 EIP 所指向的命令。

如果我们输入的字符串长度超过 16B，例如输入“hello world! AAAAAAAAAA”，则当执行完 gets(name)之后，堆栈的情况如图 11.5 所示。

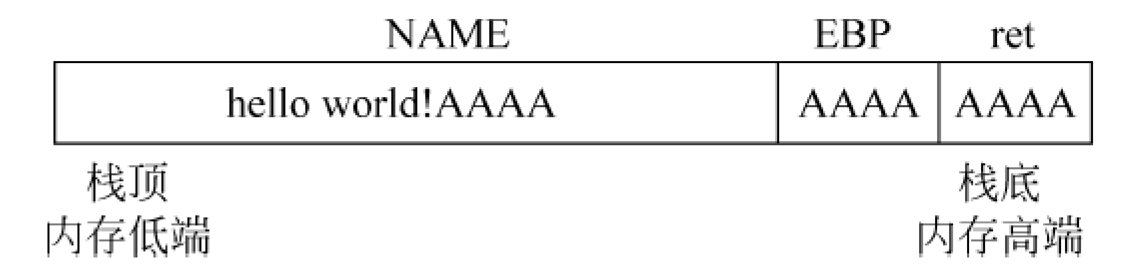


图 11.5 缓冲区溢出状态

由于输入的字符串太长，name 数组容纳不下，只好向堆栈的底部方向继续写“A”。这些“A”覆盖了堆栈中原有的元素，从图 11.5 可以看出，EBP、ret 都已经被“A”覆盖了。从 main 返回时，就必然会把“AAAA”的 ASCII 码 0x41414141 视作返回地址，CPU 会试图执

行 0x41414141 处的指令，结果出现难以预料的后果，这样就产生了一次堆栈溢出。假如使用的操作系统为 Win9X 的话，会得到那个经典的"该程序执行了非法操作"的对话框。

产生上述溢出的原因是由于堆栈是由内存的高地址向低地址方向增长，而数组变量是从内存低地址向高地址方向增长，这时如果没有对数组的访问进行越界检查和限制，通过向程序的数组缓冲区写入超过其长度的内容，就造成缓冲区溢出。

2. 覆盖堆栈中变量的内容

我们通过一个实例了解缓冲区溢出覆盖堆栈中变量引起的严重后果。

```
#include<stdio.h>
#define PASSWD "1234567890"
int verify_passwd(char  * passwd)
{
  int authenticated;
  char buffer[8];
  authenticated = strcmp(passwd,PASSWD);
  strcpy(buffer,passwd);
  return authenticated;
}
main()
{
  int valid_flag = 0;
  char passwd[1024];
  while(1){
    printf("please input passwd:");
    scanf("%s",passwd);
    valid_flag = verify_passwd(passwd);
    if(valid_flag)
      printf("incorrect passwd!\n\n");
    else
    {
      printf("Correct password!\n");
      break;
    }
  }
}
```

程序运行后要求用户输入密码，用户输入的密码与宏定义中的密码"1234567890"比较，如果密码错误，提示验证错误，并提示用户重新输入；如果密码正确，提示正确，程序退出。

按照程序设计思路，只有输入了正确的密码"1234567890"之后才能通过验证。但是由于程序存在缓冲区溢出漏洞，在某些情况下，即使输入错误的密码，也会认证通过。这段程序中的漏洞在于 verify_passwd()函数中的 strcpy(buffer，passwd)调用，strcpy 将用户输入的数据原封不动地复制到 verify_passwd 函数的局部数组 char buffer[8]中，但用户输入的字符串可能大于 8 个字符，当用户输入大于缓冲区尺寸时，缓冲区就会溢出。

函数 verify_passwd()里申请了两个局部变量：int authenticated 和 char buffer[8]，当 verify_passwd 被调用时，系统会给它分配一片连续的内存空间，这两个变量就分布在那里，用户输入的字符串将复制进 buffer[8]，authenticated 变量实际上是一个标志变量，当其值

为非 0 时,程序进入错误重输的流程;值为 0 时,进入密码正确的流程。

字符串数据最后都有作为结束标志的 NULL(0),当我们输入的字符超过 7 个,那么超出的部分将破坏与它紧邻着的 authenticated 变量的内容。例如当我们输入包含 8 个字符的错误密码 qqqqqqqq,那么 buffer[8]所拥有的 8B 将全部被 q 的 ASCII 码 0x71 填满,而字符串的结束标识 NULL 刚好写入 authenticated 变量,而且值为 0x0000000。

函数返回,main 函数看到 authenticated 是 0,就会认为密码正确,这样我们就用错误的密码得到了正确密码的运行效果。

3. 执行攻击代码

发生缓冲区溢出时,并不一定产生攻击,一次普通的缓冲区溢出使程序试图执行一处非法地址或称为错误(即不存在)的指令,这将作为非法操作而被系统中止。但是一次精心编写的缓冲区溢出程序就不同了,它在溢出的缓冲区中写入攻击者设计的可执行代码(称为 shellcode),再覆盖返回地址的内容,使它指向缓冲区可执行代码的开始,这样就可以运行攻击者的代码。如果 shellcode 执行一个 shell 时,那么这个 shell 将获得堆栈溢出程序相同的权限,即通过 shell 权限可以执行高级的命令。更为严重的是,如果这个溢出程序属于管理员用户的话,攻击者将获得一个具有管理员权限的 shell。

缓冲区溢出攻击需要由以下 3 部分组成。

(1) 在目标程序中植入攻击代码(shellcode);

(2) 实际复制进入需要溢出的缓冲区并破坏邻近区域的数据结构(通过输入超常的字符串作为输入参数来达到溢出,且溢出后返回地址指向 shellcode 的开始地址);

(3) 控制劫持(获得 shell)从而执行攻击代码(Shellcode)。

至于如何设计溢出字符串使其执行 shellcode 及如何编写 shellcode 获得 shell,请参考相关书籍,本文不作深入讨论。

4. 防范及检测方法

面对缓冲区溢出攻击的挑战,常见的防范措施主要有如下几种。

1) 安全编码

安全编码首先尽量选用自带边界检查的语言(如 Per、Python 和 Java 等)来进行程序开发;其次在使用像 C 这类开发语言时要做到尽量调用安全的库函数,对不安全的调用进行必要的边界检查。编写正确代码,除了人为注意外,还可以借助现有的一些工具,例如,使用源代码扫描工具 PurifyPlus 可以帮助发现程序中可能导致缓冲区溢出的部分。

2) 数组边界检查

只要数组的使用被严格限制在其边界之内,就不会出现缓冲区溢出问题。为了实现数组边界检查,就需要对所有对数组的读写操作进行检查,以确保对数组的操作在正确的范围内进行。

3) 返回地址

返回地址在缓冲区溢出攻击中扮演了极其重要的角色,攻击者需要借助对它的修改来使程序转向选定的库函数或者向着预先植入的恶意代码方向去执行。通过在系统的其他地方对正确的返回地址进行备份,在返回前对两者进行核对,可以成功地阻断攻击。

4）了解系统常见进程

要求用户对系统的各种正常进程有个大体的了解。用户通过了解系统中正在运行的进程，可以及时发现可疑进程，终止其运行，从而降低遭受攻击的风险。

5）及时打补丁或升级

经常关注网上公布的补丁和软件升级信息，这对用户来说是一种简单有效的防范措施。

11.2　格式化字符串漏洞

格式化字符串漏洞产生于数据输出函数（printf 等）对输出格式解析的缺陷，以 printf 函数为例：

```
int printf(const char * format,arg1,arg2,…);
```

format 的内容可能为%s、%d、%p、%x、%n…将数据格式化后输出，这种函数的问题在于 printf 函数不能确定数据参数 arg1、arg2…究竟在什么地方结束，也就是它不知道参数的个数。printf 只会根据 format 中的打印格式的数目，依次打印堆栈中参数 format 后面地址的内容。

【例 11-1】　分析下面代码的执行情况。

```
int a = 44,b = 77;
printf("a = %d,b = %d\n",a,b);
printf("a = %d,b = %d\n");
```

上述代码中第一个 printf 函数调用是正确的，输出为“a=44,b=77”，第二个调用缺少输出数据的参数列表，可编译观察一下上述代码执行的结果，上述代码执行并不会引起错误，在作者的运行环境下，得到的输出结果是“a=4218928,b=44”，下面对结果进行分析。

第一次调用 printf 函数时，函数的 3 个参数按逆序，即 b、a、“a=%d,b=%d\n”入栈，如图 11.6 所示。第二次调用 printf 函数时，由于参数中缺少输出数据列表部分，故只压入了格式控制符参数，此时堆栈状态如图 11.7 所示。虽然函数调用没有给出输出数据列表，但系统仍按格式控制符所指明的方式输出栈中紧随其后的两个 DWORD 类型值，4218928 是指向格式控制符“a=%d,b=%d\n”的指针，44 是残留下来的变量 a 的值。

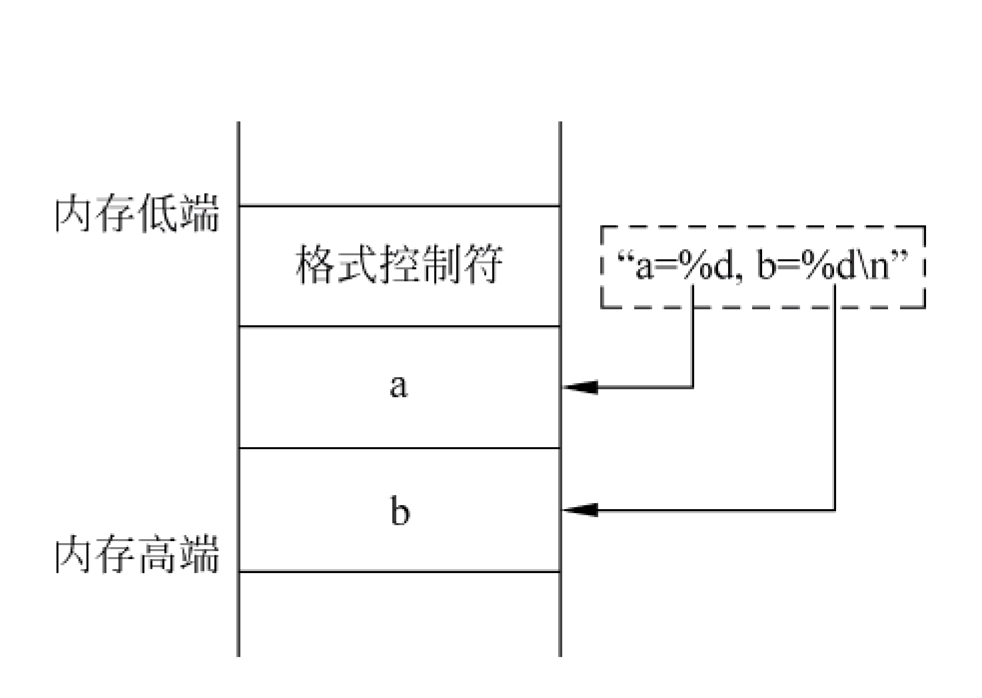


图 11.6　printf 函数调用时的内存布局

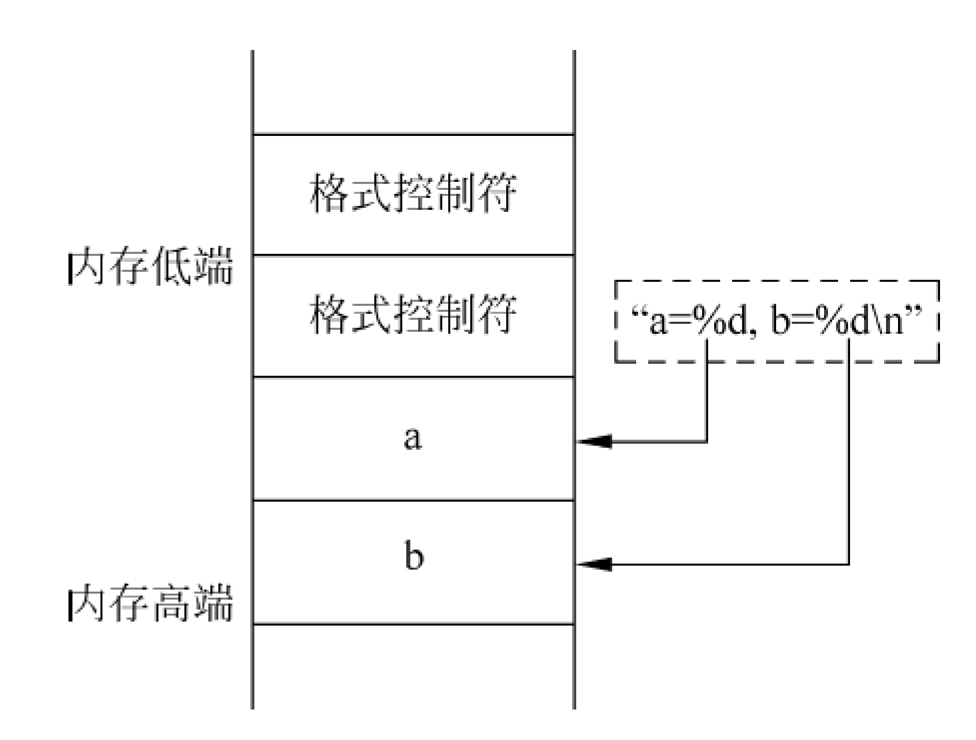


图 11.7　格式化漏洞原理

上面的例子说明了 printf 函数的设计缺陷，该缺陷可以被攻击者利用。在使用这些函数的时候，如果程序员疏忽，没有指定格式字符串参数而直接输出字符串内容，就会导致格式化字符串漏洞的发生。这种误用的问题在于来自用户的输出字符串可能包含格式化字符串，例如%s、%d 等，而实际的函数调用并没有提供任何与这些格式字符串相对应的参数，因此被调用函数的格式处理代码会试图从栈上获取其他数据，从而引发漏洞。例如在 printf(buffer)中，buffer 中的数据是"%p%p%p%p"，就可能会打印出诸如以下结果：

```
0xbffffb18  0xbffffaf8  0x80482460  0x4200aec8
```

这是因为在代码中我们没有指定对应的参数，printf 函数就会按地址格式打印出其后栈中的其他数据。如果 buffer 中有相当多的%p，就可能会打印出栈中的一些重要地址（如函数的返回地址等）。如果精心构造输入数据，还可以将存放在任意内存地址的数据显示出来。

格式化字符串漏洞除了可被利用来显示重要数据信息外，更危险的是它还可被用来向内存中写入指定的数据。"%n"就是 printf 类函数提供的用来将打印出的数据长度写入一个整型变量中的格式化字符串，例如以下这段代码：

```
int i = 0;
printf("hello % n",&i);
```

执行代码后，此时整型变量 i 的值应该被赋为 printf 函数所打印出的字符个数，即为 5。类似地，如果"%n"存在于前面那段有漏洞代码的 buffer 中，相应的内存地址就可被写入数据。如果函数的返回地址恰好被写入数据覆盖，那么程序的执行流程就被改变了。

在 C/C++语言中，可能存在与 printf 类似格式化字符串漏洞的函数还有 Snpfintf()、sprint()、fprintf()、vprintf()、vfprintf()、vsnprintf()、syslog()、vsprint()、setproctitle()等。

利用格式化字符串漏洞，可以任意读写程序内存空间中的数据信息，其可能造成的危害非常严重，因此此类漏洞是软件安全漏洞挖掘的一个重要目标。

11.3 整数溢出漏洞

整数在计算机系统中可被分为无符号整数和有符号整数两类，其中有符号正整数的最高位为 0，有符号负整数的最高位为 1；而无符号整数则没有这些限制。8 位（布尔、单字节字符等）、16 位（短整型、unicode 等）、32 位（整型、长整型）和 64 位（int64）等都是计算机系统中常见的整数类型。每种类型的整数所能表示的数值大小是有一定范围的，当对整数进行计算或转化时，如果运算的结果超出该类型的整数所能表示的范围，此时整数溢出就会产生。整数溢出主要可分为以下 3 种情况。

(1) 存储溢出：是由使用不同的数据类型来存储整数造成的。当在一个较小的整数变量存储区域中放入一个较大的整数变量，就会造成截断多余位只保留较小变量所能存储的位数的结果。

(2) 计算溢出：在计算整型变量的过程中，对其所能表示的边界范围没考虑到，从而导致计算得到的数值超出了其最大存储空间。

(3) 符号问题：有符号整数和无符号整数是整数的两种类型，一般要求数据的长度变量都要使用无符号整数，如果符号问题被程序员所忽略，那么意想不到的情况就可能发生在对数值的边界进行安全检查时。

几乎所有整数溢出漏洞都因为算术运算(如加、乘等)中涉及未审核的用户可修改的数值，而这些运算的潜在溢出结果值都被用于关键操作中(如内存的分配、复制等)。例如如下所示代码：

```
char * integer_overflow(int *data,unsigned int len)
{
  unsigned int size = len + 1;
  char * buf = (Char *)malloc(size);
  if(!buf) return NULL;
  memcpy(buf,data,len);
  buf[len] = '\0';
  return buf;
}
```

用户输入的整型数据会被该函数复制到新的缓冲区中，并且结尾符“\0”会在其最后被写入。如果 0xFFFFFFFF 被攻击者作为参数 len 的值传入，那么整数溢出就会在计算 size 时发生，随后 malloc 函数就会根据 size 的值分配大小为 0 的内存块，而紧接着执行 memcpy 时就会发生堆溢出。通常情况下，单独利用整数溢出来实现攻击目的是不太可能的，整数溢出大多是被用来绕过程序中存在的条件检测等限制，进而引发其他漏洞来完成攻击，正如上面例子所示的缓冲区溢出就是利用整数溢出引发的。

11.4　应用系统安全漏洞发掘方法

软件安全漏洞发掘是一个涉及诸多技术的复杂工程，主要可分为对已知漏洞的检测和对未知漏洞的发掘。对已知漏洞的检测主要是通过安全扫描技术，来发现软件系统是否存在已公布的安全漏洞。而对未知漏洞发掘的主要目的在于发现软件系统中可能存在但尚未被发现的安全漏洞。依据不同的划分标准，可将软件安全漏洞发掘方法划分为不同类型。根据漏洞发掘的自动化程度，可分为手工分析、半自动或自动化分析；根据软件源代码的开放性，可分为白盒分析、黑盒分析和灰盒分析 3 类。白盒分析是指拥有被测程序的源代码和设计文档等资料；黑盒分析是指没有源代码而只能运行目标软件观察其输出结果来进行分析；灰盒分析介于二者之间，通过逆向工程获得目标软件的汇编代码来进行分析。根据目标软件的运行状态，可分为静态分析和动态分析。静态分析方法的使用无须运行程序，而动态分析方法则需要运行程序进行动态调试分析。针对具体的测试对象可以分别采用不同的发掘方法，也可多种分析方法结合使用。下面将重点对静态分析方法和动态分析方法进行一些分析和总结。

1. 静态分析方法

静态分析方法获取程序的各种调用信息和动态特性是通过静态分析程序代码来实现的，并不需要执行程序。静态分析方法与动态分析方法相比，通常所花费的检测时间更少。

常见的静态分析方法主要有词法语法分析、模式匹配分析、数据流分析、补丁比较分析和模型化分析方法等。

模式匹配通过依据一定的规则或模式在程序源代码或反汇编二进制代码得到的汇编代码中发掘软件中的漏洞。软件中需要重点检查的区域有：没有执行或执行了不正确的边界检查的函数、使用用户数据作为参数或以字符串格式进行传参的函数、强制通过格式化字符串进行边界检查的函数、使用循环操作获得用户输入的程序、原始的字符复制操作、在用户缓冲区内进行的指针运算等。词法语法分析方法在与C编译器创建的抽象语法表示类似的程序语法的基础上进行分析。数据流分析方法判断程序中的漏洞是依据数据在程序运行时的流向来进行的。此外还有通过标记注释的方法来研究分析程序源代码，以分析查找漏洞。对静态分析方法的研究日趋完善和成熟，此方面的许多研究成果都成了漏洞挖掘的经典技术和方法。

2. 动态分析法

动态分析技术一般是通过在调试器中加载目标程序并观察程序在运行过程中的状态(如寄存器内容、内存使用状况等)以发现潜在的漏洞。代码流和数据流是它通常的两个着手点：通过在程序中设置断点来对目标程序的代码执行进行动态跟踪，以检测出有问题的函数调用；动态分析数据流，通过构造特殊输入数据来触发程序的潜在错误并对结果进行分析。动态分析通常需要有调试器工具的支持，常用的动态调试器有OllyDbg、WinDbg等。故障注入分析法、输入追踪法和堆栈比较法等都是比较常见的动态分析方法。

11.5 Web应用安全

随着互联网相关技术的飞速发展，Web应用程序凭借其交互性和易用性风靡世界。但是Web应用程序的开发周期较短，开发技术更新换代快而且易于入门，这就造成了在开发过程中，开发人员的安全意识相对淡薄，对安全方面的重视不够，只是以实现功能、界面美观为主要目的。另外，Web应用程序具有开放性，用户访问量很大，这加大了被攻击的概率。OWASP组织总结的Web应用漏洞已有上百种之多。很多黑客通过Web应用漏洞窃取金钱、隐私、扰乱正常的业务执行。据调查显示，网络上超过70%的攻击来自于Web应用层的漏洞。

11.5.1 Web应用基础

1. HTTP协议模型

HTTP协议是一种用于客户端到Web服务器之间请求和应答的通信协议。HTTP协议在网络体系协议中处于应用层协议，主要由HTTP请求和HTTP响应组成，是典型的B/S模型。HTTP是一个无状态的协议。HTTP协议总是由客户端发起请求，服务器经过处理请求后回传响应，服务器端只能被动地接受请求，如图11.8所示。一次HTTP操作称为一个事务，其工作过程可分为以下4步。



图11.8 HTTP协议应用模式

(1) 用户输入一个网址,浏览器对网址进行解析,提取出使用的协议版本、网址的域名、所需要获取目标的路径,可能还带有一些参数。

(2) 浏览器利用第一步解析的信息构造 HTTP 请求,并将请求发送到服务器端。

(3) 服务器端收到请求后就处理请求,其中有可能还要重定向,服务器处理完请求后产生了一个响应。

(4) 服务器将响应以 HTML 文档的形式发送给浏览器,浏览器解析 HTTP 的响应头,判断返回码是否成功返回所需要的网页,浏览器渲染 HTML 文档,显示界面。

最早的 WWW 服务器上包含的都是静态文件,有 HTML 网页、音频、图片、文档等。服务器一直在等待客户端的请求,当收到一个请求时,服务器就处理请求,寻找到用户请求的静态文件,将文件发送给客户端。

如今的 Web 服务器主要提供动态生成的内容,动态内容由在服务器上的脚本产生,Web 应用处理用户提交的各种输入,通过带查询字符串的 URL 和 POST 方法提交数据,服务器根据用户参数产生动态的内容,将内容返回给用户。

2. Web 应用体系结构

Web 应用指采用浏览器/服务器(Browser/Server,B/S)架构、通过超文本传输协议(HyperText Transfer Protocol,HTTP)或以安全为目标的 HTTP 通道(Hypertext Transfer Protocol over Secure SocketLayer,HTTPS)协议提供访问的各种应用服务统称,如图 11.9 所示,是一个典型的 Web 应用架构。Web 应用通常就是我们所说的网站,包含文字、图片、视频、音频、CSS 文件、Javascript 文件等静态文件,还包括一些根据用户请求动态生成的动态页面。搜索引擎就是一个典型的 Web 应用,它根据用户提交的关键字动态生成不同的页面。

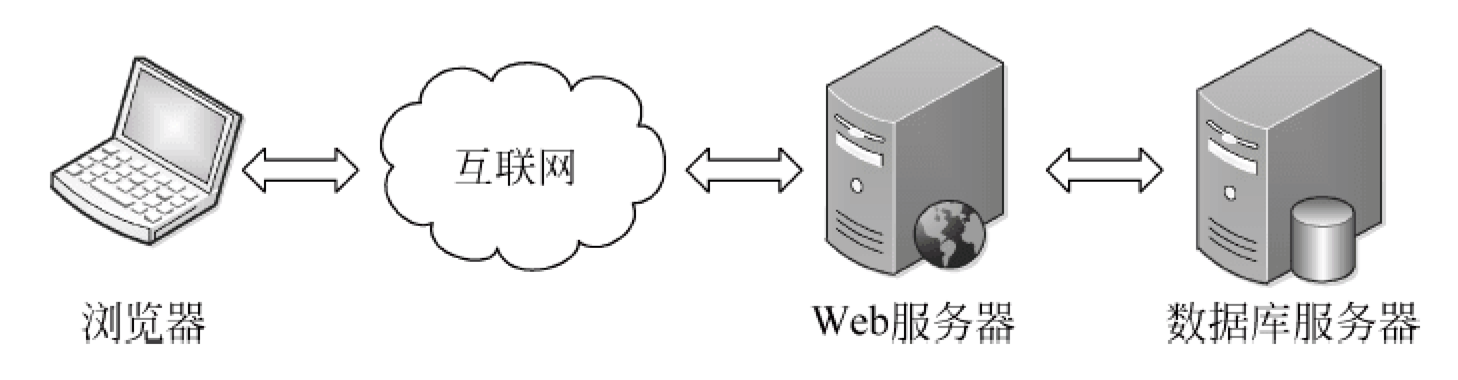


图 11.9　Web 应用架构

常见的 Web 应用程序都是基于模型-视图-控制器(MVC)设计的,通常由表示层、应用层、数据层组成。

(1) 表示层:表示层直接与用户进行交互,获取用户的输入,验证输入的合法性,并传递给处在服务器端的应用层,应用层将处理的结果返回给表示层,客户端浏览器以 HTML 的形式显示处理的结果。

(2) 应用层:应用层通常也被称为业务层,它是表示层和数据层的桥梁,位于 Web 服务器端,它处理用户提交上来的请求,控制将用户的请求发送给每一层的业务逻辑,另外还调用数据层的接口,获取数据库的数据,产生 HTTP 的响应。

(3) 数据层:数据层 Web 应用程序与数据库的交互,连接和访问数据库,通过 SQL 语句向用户提交数据处理请求,查询、更新、删除、修改相关数据,并将数据处理结果返回给 Web 应用服务器,再由 Web 应用服务器返回给客户端。

11.5.2 Web 应用漏洞

Web 应用程序漏洞通常被认为是各种编程语言(PHP、JSP、Python 等)开发的 Web Application/Web Service 中所存在的安全漏洞。如果这些安全漏洞被发掘,可能被利用来攻击 Web 应用、篡改网页信息、盗取数据库信息,甚至有可能提权,获取服务器的管理权限。Web 应用程序漏洞包括 SQL 注入漏洞、XSS 漏洞、信息泄漏漏洞、跨网站请求漏洞、命令注入漏洞等。

OWASP(Open Web Appliaction Security Project)组织的 Top10 项目评出的 Web 应用十大应用漏洞显示,SQL 注入攻击、跨站脚本攻击的威胁最大,分别占到了 16%、19%的比例。本文仅对这两类攻击的原理进行分析。

11.5.3 SQL 注入漏洞及防御机制

1. 概述

在 Web 应用中,一个 SQL 查询过程一般是 Web 服务器根据用户的查询请求(用户参数)拼接成 SQL 语句,交由数据库服务器执行,Web 服务器根据 SQL 语句执行结果构建 Web 动态页面并发送给客户。但是当用户提交恶意请求,而 Web 应用程序如果对用户提交的参数未做过滤就直接构造 SQL 语句,就会使得数据库服务器执行攻击者精心构建的 SQL 语句或语句序列,从而达到破坏数据库、获取机密数据,甚至可能盗取数据库服务器的管理员权限的目的,这种攻击称为 SQL 注入(SQL Injection)攻击。

一般而言,SQL 注入攻击对于各种关系型数据库(如 DB2、Oracle、Sybase、MySQL、Microsoft SQL Server 等)均有效,但是由于各个数据库的 SQL 语法不尽相同,其攻击语句可能会有所差别。

下面以登录验证为例,说明 SQL 注入攻击的实现方法。在 Web 应用程序的登录界面中需要输入用户名、口令信息,如图 11.10 所示,这些信息以参数的形式传送到 Web 服务器,Web 服务器根据用户参数构建 SQL 语句,交由数据库服务器执行查询,根据查询结果验证当前用户是否为合法用户,构建的 SQL 语句可表述为:

```
SELECT ID
FROM USERS
WHERE UAERNAME = 'chenping' AND PASSWD = '123456';
```

对于攻击者来说,为了绕过用户身份检查,可在登录界面的用户名表单中输入"test' or 1=1--",这样在服务器端根据用户输入构成的 SQL 语句就为:

```
SELECT ID
FROM USERS
WHERE UAERNAME = 'test' or 1 = 1 -- 'AND PASSWD = '';
```

该语句进行了两个判断,只要一个条件成立,就会执行成功,而 1=1 在逻辑判断上是恒成立的,后面的“--”表示注释,即后面所有的语句为注释语句。这样上述 SQL 语句 WHERE 条件表达式恒为真,用户不需要正确的用户名和密码也能登录系统。

如果攻击者在登录界面的用户名表单中输入的信息为"test' or 1=1; drop table users ;--",



图 11.10　登录界面

不仅可以绕过身份检查，而且还删除了数据库中的表。同理通过在输入参数中构建 SQL 语法还可以执行查询、插入和更新数据库中的数据等危险操作。

(1) ;union select sum(username)from users：从 users 表中查询出 username 的个数。

(2) ;insert into users values(666,'attacker,'foobar',0xffff)：在 users 表中插入值。

(3) ;union select@@version,1,1,1：查询数据库的版本。

(4) ;exec master..xp_cmdshell 'dir'：通过 xp_cmdshell 来执行 dir 命令。

2. 防御

SQL 注入漏洞会导致攻击者可以直接探测数据库，导致的危害非常大，轻则泄露数据，重则失去系统的管理员权限。所以非常有必要加紧防范，再怎么重视也不过分，其中主要的防御措施有：

(1) 在服务端正式处理之前对提交数据的合法性进行检查。

(2) 封装客户端提交信息。

(3) 替换或删除敏感字符/字符串。

(4) 屏蔽出错信息。

(5) 不要用字串连接建立 SQL 查询，而使用 SQL 变量，因为变量不是可以执行的脚本。

(6) 目录最小化权限设置，给静态网页目录和动态网页目录分别设置不同权限，尽量不给写目录权限。

(7) 修改或者去掉 Web 服务器上默认的一些危险命令，例如 ftp、cmd、wscript 等，需要时再复制到相应目录。

(8) 数据敏感信息非常规加密，通常在程序中对口令等敏感信息加密都是采用 MD5 函数进行加密，即密文＝MD5(明文)，本书推荐在原来的加密的基础上增加一些非常规的方式，即在 MD5 加密的基础上附带一些值，如密文＝MD5(MD5(明文)＋123456)。

11.5.4　XSS 注入漏洞及防御机制

1. 概述

跨站脚本漏洞(Cross-site Scripting)是一种针对 Web 应用程序的安全漏洞。它一般指

的是利用网页开发时留下的安全漏洞,使用特殊的方法将恶意 Javascript 代码注入到网页中,用户在不知不觉中加载并执行攻击者构造的恶意 Javascript 代码。当攻击成功后,攻击者可能执行下列操作:挂马、钓鱼、获取私密网页内容、劫持用户 Web 行为、盗取会话和 cookie 等。大量的网站曾经遭受 XSS 漏洞攻击或被发现此类漏洞,如 Twitter、Facebook、MySpace、新浪微博和百度贴吧。

构成跨站脚本漏洞的主要原因是很多网站提供了用户交互的页面,如检索、留言本、论坛等,凡是能够提供信息输入,同时又会将提交信息作为网站页面内容输出的地方都可能存在跨站脚本漏洞,服务器程序对输入信息检查不严格导致了脚本嵌入的可能。

2. 分类

XSS 漏洞有 3 类:反射型 XSS(也叫非持久型 XSS 漏洞)、存储型 XSS 和 DOM 型 XSS。

1) 反射型 XSS

反射型 XSS 是最常用,也是使用得最广泛的一种攻击方式,它通过给别人发送带有恶意脚本代码参数的 URL,当 URL 地址被打开时,特有的恶意代码参数被 HTML 解析、执行。它的特点是非持久化,必须由用户单击带有特定参数的链接才能引起。例如有以下 index. php 页面:

```
<?php
 $ username  =  $ _GET["name"];
echo "< p >欢迎您, ". $ username."!</p >";
?>
```

正常情况下,用户会在 URL 中提交参数 name 的值为自己的姓名,然后该数据内容会通过以上代码在页面中展示,如用户提交姓名为“张三”,完整的 URL 地址如下:

```
http://localhost/test.php?name = 张三
```

在浏览器中访问时,会显示如图 11.11 所示的内容。

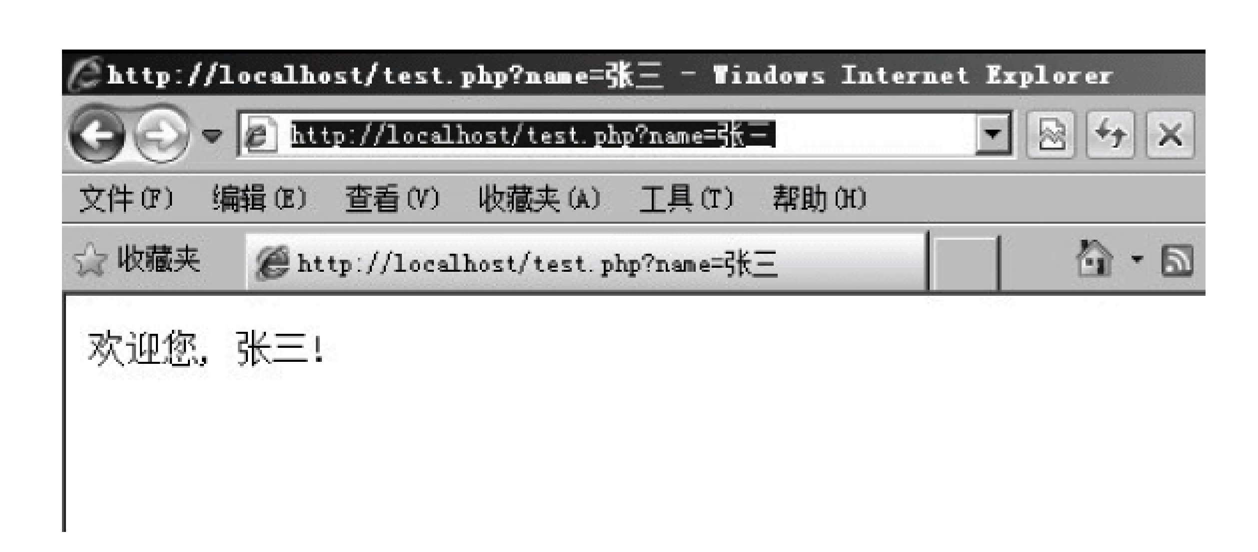


图 11.11 正常访问界面

此时,因为用户输入的数据信息为正常数据信息,经过脚本处理以后页面反馈的源码内容为＜p＞欢迎您,张三!＜/p＞。但是如果用户提交的数据中包含有可能被浏览器执行的代码的话,会是一种什么情况呢?我们继续提交 name 的值为＜script＞alert(/我的名字是张三/)＜/script＞,即完整的 URL 地址为 http://localhost/test. php? name=＜script＞alert(/我的名字是张三/)＜/script＞。

在浏览器中访问时,我们发现会有弹窗提示,如图 11.12 所示。

那么此时页面的源码又是什么情况呢?



图 11.12　运行脚本界面

源码变成了"<p>欢迎您,<script>alert(/我的名字是张三/)</script>! </p>",我们从源代码中发现,用户输入的数据中,<script>与</script>标签中的代码被浏览器执行了,而这并不是网页脚本程序想要的结果。这个例子正是最简单的一种 XSS 跨站脚本攻击的形式,称为反射型 XSS。

2) 存储型 XSS

存储型 XSS 又称为永久性 XSS,它的危害更大,它与反射型 XSS 漏洞的区别在于,它提交的 XSS 代码会存储在服务器中,有可能存在数据库中,也有可能存在文件系统中,当其他用户请求带有这个注入 XSS 代码的网页时就会下载并执行它。最典型的例子就是留言板 XSS,用户提交一条包含 XSS 代码的留言存储到数据库,目标用户查看留言板时,那些留言的内容会从数据库查询出来并显示,浏览器发现有 XSS 代码,就当作 HTML 和 Javascript 代码解析执行,于是就触发了 XSS 攻击,存储型 XSS 的攻击是很隐蔽的,不容易通过手工查询发现,需要采用自动化的 Web 应用漏洞扫描器。

3) DOM 型 XSS

DOM 型 XSS 和反射型 XSS、存储型 XSS 的差别在于:DOM 型 XSS 的 XSS 代码并不需要服务器解析响应的直接参与,触发 XSS 靠的就是浏览器端的 DOM 解析,可以认为完全是客户端的事情。举例如下:

```
<script>
eval(location.hash.substr(1));
</script>
```

这就是一个 DOM 型 XSS 漏洞,触发方式为提交请求 http://www.foo.com/xssme.html#alert(1),这个 url# 后面的内容不会被发送到服务端,仅仅是在客户段被接收并解析执行。在 Javascript 中有很多这种输入点可以注入恶意脚本。

3. XSS 漏洞的防范措施

跨站脚本攻击相对于其他网络漏洞攻击而言显得更为隐蔽,也更难防御,没有一劳永逸的解决办法。XSS 漏洞是在用户和 Web 应用程序交互的过程中产生的,既有 Web 应用程序本身的问题也有客户端用户的问题。主要应将防御的重心放在 Web 应用程序的编程上,但是用户良好的使用习惯也能够尽量避免 XSS 攻击。所以我们分两个方面入手防御跨站

脚本攻击,程序开发者在编程过程中要进行输入验证,用户在浏览网页时也应采取相应安全措施,程序开发过程中应该采取的措施如下。

(1) 过滤用户提交数据中的代码。这种方法的实施过程比较复杂,不仅仅需要考虑<script>、</script>标签,各种可能的XSS攻击载体都要考虑进来。将所有非法输入保存成黑名单方式过滤数据是不太可能实现的,更好的方式就是只接收合法的数据。

(2) 对用户输入的数据或基于用户输入数据而生成的输出数据进行编码。一般而言,编码是很简单也是很有效的防范XSS脚本的方法,因为它不需要区别合法字符和非法字符。这么做的缺陷是,对所有不可信数据编码非常浪费系统资源,可能影响到Web服务器的性能。

(3) 对表单输入域输入字符的长度加以限制。对于一些可能受到攻击的表单输入域,可以限制其输入字符的长度。

(4) 禁止上传Flash。文件利用Flash文件进行跨站脚本攻击难于防范,如果不能确定上传的Flash文件是否安全,那就干脆禁止用户上传Flash文件,以彻底阻断Flash跨站攻击的途径。

(5) 检查cookie信息。许多Web应用程序利用cookie来管理通信状态,并存储与用户相关的信息。开发人员必须对cookie信息进行严格的检查和过滤之后才能将其插入HTML文档。

用户的应对措施如下。

(1) 慎重点击不可信的链接,只点击一些可信链接、可信网站。有时候XSS攻击会在打开电子邮件、阅读留言板、打开附件、阅读论坛时不经意发生。

(2) 提高浏览器的安全等级,及时将浏览器更新到最新版本,将浏览器的安全级别设置为高,同时禁用一些不需要运行的脚本。

(3) 要在不同的Web应用程序中使用不同的用户名和密码。

11.6 小　　结

由于应用程序在开发中不可避免会存在无意的缺陷或故意留下的安全漏洞,如何在应用系统开发过程中避免漏洞,如何发现、检测应用系统存在的漏洞,是保障应用系统及其运行平台安全的重要方面。本章主要讨论了常见的缓冲区溢出漏洞、格式化字符串漏洞、整数溢出漏洞的原理和预防、检测方法,并针对目前应用广泛的Web系统的安全性进行了讨论,重点介绍了SQL注入攻击和跨站脚本攻击的原理和防范措施。

习　　题

一、填空题

1. ________攻击一直是网络攻击事件中用得最多的一种攻击方式。
2. “冲击波”和“震荡波”都是利用________漏洞进行攻击的。

3. 格式化字符串漏洞的根源在于________。

4. 整数溢出主要分三种情况：________、________、________。

5. Web 应用架构包括以下组成部分：________、________、________。

6. 跨站脚本漏洞有 3 类：________、________、________。

二、简答题

1. 什么是缓冲区溢出攻击？

2. 缓冲区溢出的原因是什么？如何防范？

3. 请描述在函数调用时，对堆栈的操作步骤。

4. 查阅相关资料，谈谈你对白盒测试、黑盒测试、灰盒测试的看法。

5. 请谈谈对 SQL 注入攻击的认识。

6. 请谈谈对跨站脚本攻击的认识。

第 12 章　信息系统安全新技术

随着信息技术的快速发展，信息系统在各行各业得到了越来越广泛的应用，一些基于云计算技术、物联网技术和移动互联网技术的新型信息系统大量出现，并在国家、企业和个人中迅速普及。这些新型信息系统的使用在给组织和个人的工作、生活带来高效和便利的同时，也引入了新的信息安全问题，需要针对各自特点研究其信息安全防护技术，提高其信息安全防护能力。

本章对信息系统安全的新技术进行了简要介绍，12.1 节介绍云计算系统的基本概念、面临的主要安全威胁和主要应对策略，12.2 节讲述物联网的概念、体系结构和安全新特征，12.3 节介绍移动互联网的发展现状、安全性方面的特点以及主要安全机制，12.4 节对本章的内容进行小结。

12.1　云计算信息系统及其安全技术

当前，云计算已经成为通信、IT 界关注的重点，各方均看好其市场发展前景。云计算本质上是传统电信 IDC 增值业务的延伸和扩展，通过互联网对用户提供 IT 基础资源（包括计算、存储、网络、软件等）的按需租用，能够降低用户的 IT 运维成本，使得用户可以专注于自身业务。由于云计算的独特优势，欧美等国家政府均大力推广使用此项技术，云计算的广泛普及对工业化和信息化的快速融合及国民经济发展均有促进作用。

云计算具有按需服务、宽带接入、虚拟化资源池、快速弹性架构、可测量的服务和多租户等特点，不但对传统的安全提出了挑战，同时也为 IT 系统引入了新的风险。因此，在云计算快速推进、广泛普及的同时，有必要重点对云安全技术进行研究，在云中引入更强大的安全措施，否则，云的特性以及云提供的服务不仅无法有效利用，而且还可能给国家、企业、个人用户带来严重的安全威胁。

一般来说，云计算的安全包括两个不同的研究方向：

(1) 云安全，即保护云计算系统本身的安全。

(2) 安全云，属于云计算应用范畴，即利用云的特性，将云作为一种安全服务提供给第三方。

随后章节将对云计算安全的第一个层次进行阐述。

1. 云计算模型

我们使用 NIST（美国国家标准与技术研究院）给出的云计算模型。简要地说，云模型可以解读为 1 个平台、2 个支付方案（按使用量收费和按服务收费）、3 个交付模式（IaaS、PaaS、SaaS）、4 个部署模式（私有云、公有云、社区云、混合云）、5 个关键特性（基础资源租用、按需弹性使用、透明资源访问、自助业务部署、开放公众服务），详见图 12.1。

NIST 制订的《云计算工作定义》归纳了云计算的 3 种交付模式，即基础设施即服务

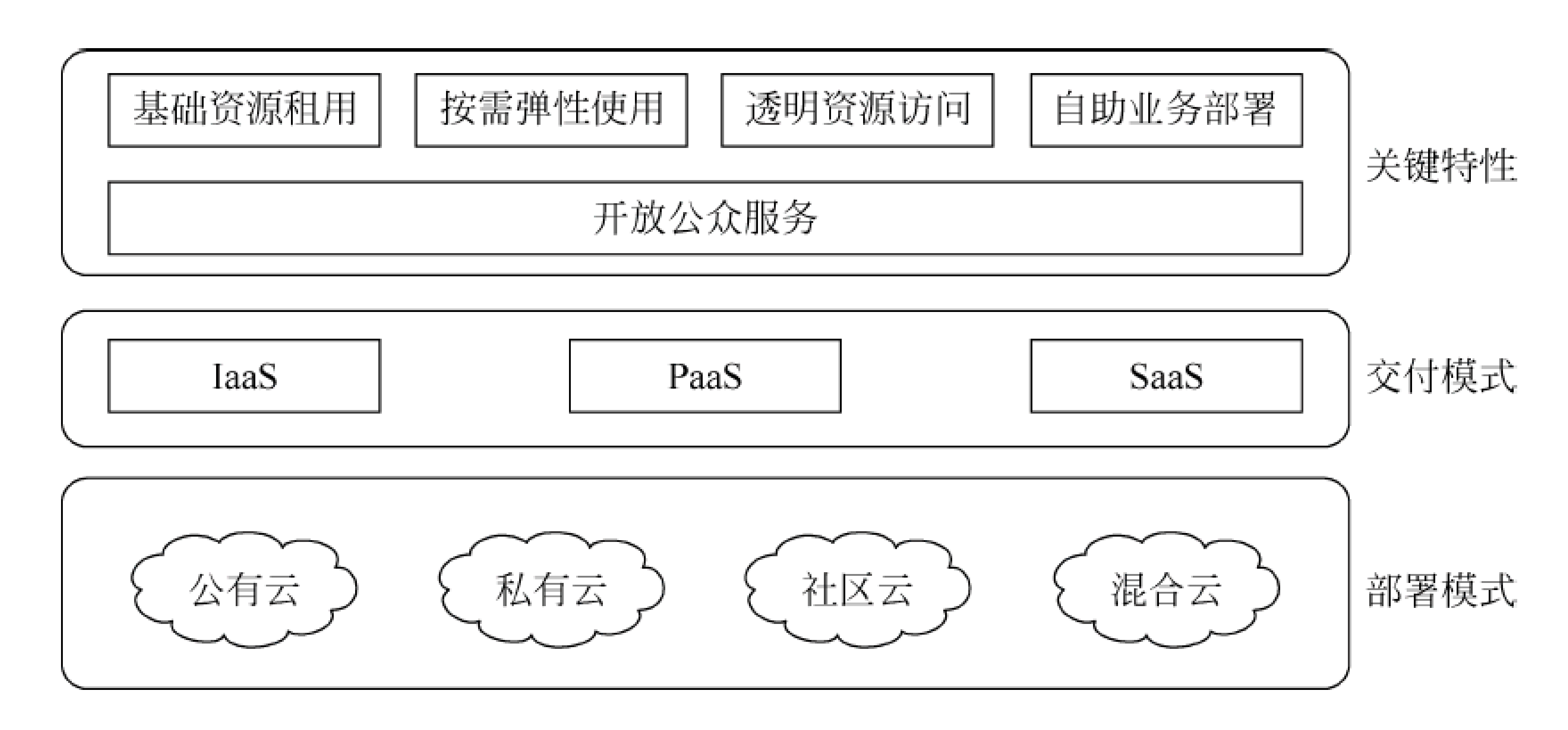


图 12.1　云计算模型结构图

(Infrastructure as a Service,IaaS)、平台即服务(Platform as a Service,PaaS)、软件即服务(Software as a Service,SaaS)。

下面从集成特色功能、复杂性、扩展性以及安全性等方面对 3 种交付模型进行比较。

一般来说,SaaS 会在产品中提供强大的集成化功能,对用户而言使用简单、安全威胁较小,但可扩展能力差。

PaaS 为开发者提供在平台之上进行二次开发的能力,因此可以提供比 SaaS 更多的可扩展性,其代价是牺牲 SaaS 中为用户提供的特色功能。这种折中也会影响安全能力,虽然 PaaS 内置的安全能力不够完备,但是用户却拥有更多的灵活性去保证安全。

IaaS 几乎不提供上层应用及服务,但却有最好的可扩展性。同时 IaaS 仅提供保护基础设施自身的安全机制,要求云用户自己管理和保护操作系统、应用和内容。

2. 国内外云计算安全研究动态

云计算引发的数据和信息泄露、隐私保护和被滥用等安全问题已经引起了各国政府的高度重视。作为一种新兴的节省 IT 运维成本的服务模式,美国政府在大力推广云计算应用的同时,也在积极采取各项措施应对云计算可能产生的各种风险。

据报道,美国联邦贸易委员会(Federal Trade Commission,FTC)调查了云计算可能存在的风险,并于 2009 年 3 月对云计算的利弊举行了听证会。一些对云计算持反对意见的人,包括美国电子隐私信息中心(Electronic Privacy Information Center,EPIC)认为,由于数据管理做法的不同,消费者在云计算下个人信息的泄露风险更大。根据 ComScore Media Metrix 的统计结果表示,仅在 2008 年 9 月和 10 月,就有 3040 万用户使用了 Google 的 Docs 和 Gmail 等服务。Google 的云计算服务中存在多个安全漏洞,典型的漏洞如存在被人窃取登录信息和窥探用户电子邮件,以及通过恶意网站泄露用户个人数据的漏洞。EPIC 甚至要求 FTC 以法律明文阻止云计算的部署。同时 EPIC 要求 FTC 让 Google 将其安全政策更加透明化,将所有数据泄露和丢失向 FTC 报告,并要求 Google 专门资助云系统的隐私保护研究。

FTC 听证会的主要目的是解决以下问题:为确保云计算上的个人信息安全,哪些机构可以管辖数据交换,改变管辖权后政府能否访问这类信息?由一个机构管辖的数据中心存储个人信息与多个数据中心存储相比,哪个风险更大?由于美国当前州法律、国家法律与国际法律都没有完善的法律可依,身份信息盗窃问题日益严重,如果监管不严,数据管理服务

将遭遇崩溃。

云安全联盟(Cloud Security Alliance,CSA)成立于2009年,该组织专注于云计算的安全体系及安全标准等领域,其宗旨是为云计算环境下提供最佳的安全方案,并公开了多部涉及云安全内容的白皮书。2011年11月14日,CSA发布了第三版《云计算关键领域安全指南》,该白皮书对云安全所涉及的15大领域进行检查,其中包括支配和企业风险、信息和全过程管理、适应性与审计、eDiscovery企业内容管理平台、加密和密钥管理、应用程序安全、身份和访问管理以及事件响应等。

截至2011年11月月底,CSA和ISACA(国际信息系统审计协会)、ITU(国际电信联盟)、OWASP(开放式Web应用程序安全项目)等10个业界标准组织建立了合作关系,其企业会员已达105个。

为了支持云计算的标准化和协同工作能力、包括IBM、Sun Microsystems、VMware在内的一些知名厂商组成的团体签署了一份《开放云计算宣言》(*Open Cloud Manifest*)。文件将致力于解决云计算环境中所出现的周边安全、整合、协同工作、支配/管理和测量/监控等问题。

3. 云计算中的安全问题及应对策略

云计算作为新的服务模式,在带来了诸多好处的同时也面临着巨大的安全挑战。在云环境下,传统的安全机制将面临云架构的挑战。弹性资源分配、多租户、新的物理和逻辑架构、数据在外部甚至公众的环境中传输都需要新的安全策略。据国际权威咨询集团IDC的调查分析表明:用户将安全作为对云计算的主要顾虑,大约有75%的受访者担心安全问题。

本节将对云中的数据安全、应用安全、虚拟化安全、云服务滥用等问题及应对策略进行重点介绍。

1) 云中的数据安全

云计算环境下,用户的所有数据直接存储在云中,在需要的时候直接从云端下载使用。用户使用的软件由服务商统一部署在云端运行,软件维护由服务商来完成,当终端出现故障时,不会对用户造成影响,用户只需要更换终端,接入云服务就可以获得数据。实现上述描述的前提是,云服务商需要具备完善的数据安全机制。一般说来,保护云中数据安全,需要如下技术:

(1) 增强加密技术。增强加密是云计算系统保护数据的一种核心机制。加密提供了资源保护功能,同时密钥管理则提供了对受保护资源的访问控制。云服务商需要同时对网络中传输的数据及云系统中的静态数据进行加密,后者尤为关键。加密磁盘上的数据或生产数据库中的数据可以用来防止恶意的云服务提供商、恶意的邻居“租户”及某些类型应用的滥用。此外,一些用户可能会有如下需求,首先加密自己的数据,然后将密文发送给云服务商,客户控制并保存密钥,在需要的情况下解密数据。

在IaaS环境中,使用多种提供商和第三方工具加密静止数据非常普遍。在PaaS环境中加密静止数据一般会比较复杂,需要提供商提供的或专门定制的设备。在SaaS环境中云用户无法直接加密静止数据,需要云服务提供商为其加密。

(2) 密钥管理。对于云服务商而言,密钥必须像其他敏感数据一样进行保护。在存储、传输和备份过程中都必须保护密钥的安全,较差的密钥存储方案可能对加密的数据产生严重威胁。

同时云服务商还需要相关策略来管理密钥的存储,例如利用角色分离进行访问控制,针对某一密钥,使用实体不能是存储该密钥的实体。

丢失密钥意味着被此密钥所保护的数据面临严重安全风险,运营商必须向用户提供安全备份和安全恢复的解决方案。

(3) 数据隔离。在多租户环境下,不同用户的数据可能会混合存储。虽然云计算应用在设计时采用多种技术标注数据存储空间,防止非法访问混合数据,但是通过应用程序的漏洞,非法访问还是会发生,例如,Gmail 系统曾经出现过类似问题,某些用户可以非法获得其他用户的邮件。虽然云服务提供商会使用安全机制减少此类安全事件发生的概率,但从本质上看,如果无法实现单租户专用数据平台,这种安全威胁将无法彻底根除。

(4) 数据残留是数据在被以某种形式擦除后所残留的物理表现,存储介质被擦除后可能留有一些物理特性使数据能够被重建。由于云计算的动态分配、资源可扩展特性,某一块存储空间在短时间内可分配给多个用户,如果云服务商不能彻底清除之前用户的历史数据,则后来用户可能通过残留的数据,获取其他用户的敏感信息。因此,云服务提供商需具备相应的安全能力,无论用户的信息存放在硬盘上还是在内存中,应保证在二次分配之前彻底清除当前用户的信息,保证系统内的文件、目录和数据库记录等资源所在的存储空间被释放,或重新分配给其他云用户前完全清除。

2) 应用安全

由于云环境的灵活性、开放性以及公众可用性等特性,给应用安全带来了很大挑战。云服务商在部署应用程序时应当充分考虑未来可能引发的安全风险。对于使用云服务的用户而言,应提高安全意识,采取必要措施,保证云终端的安全。例如,用户可以在处理敏感数据的应用程序服务器之间通信时采用加密技术,以确保其机密性。云用户应定期自动更新,及时为使用云服务的应用打补丁或更新版本。

3) 虚拟化安全

虚拟化是云计算的重要特色,虚拟化技术有效加强了基础设施、平台、软件层面的扩展能力,但虚拟化技术的应用使得传统物理安全边界缺失,传统的基于安全域/安全边界的防护机制难以满足虚拟化下的多租户应用模式,用户信息安全使用户信息隔离问题在共享物理资源环境下的保护更为迫切。

Gartner 公司研究表明,60%的虚拟化服务器比物理基础设施更容易遭到攻击,不是因为虚拟化本身不安全,而在于系统配置方面。

研究发现,40%的虚拟化部署在最初的架构和规划阶段甚至都没有考虑到 IT 安全因素。Gartner 建议安全流程应该扩展到虚拟化管理程序和虚拟机监视器方面。虚拟化层可能包含嵌入的和没有发现的安全漏洞,这些安全漏洞一旦被利用,将会对云系统造成严重影响。

虚拟化技术为云计算引入的风险包括两个方面,即虚拟化软件安全和虚拟服务器的安全。虚拟化软件直接部署于裸机之上,提供能够创建、运行和销毁虚拟服务器的能力。虚拟化软件层是保证客户的虚拟机在多租户环境下相互隔离的重要层次,可以使客户在一台计算机上安全地同时运行多个操作系统,所以必须严格限制任何未经授权的用户访问虚拟化软件层。云服务提供商应建立必要的安全控制措施,限制对于 hypervisor 和其他形式的虚

拟化层次的物理及逻辑访问。

虚拟服务器位于虚拟化软件之上，虚拟服务器安全包括物理机选择、虚拟服务器安全和日常管理3方面。

虚拟操作系统管理方面的现状是大多数提供缺省安全保护的进程都未被加入，因此必须特别注意如何代替它们的功能。虚拟化技术本身引入了hypervisor和其他管理模块这些新的攻击层面，但更重要的是虚拟化对网络安全带来的严重威胁，虚拟机间通过硬件的背板而不是网络进行通信，因此，这些通信流量对标准的网络安全控制来说是不可见的，无法对它们进行监测、在线封堵，类似这些安全控制功能在虚拟化环境中都需要采用新的形式。

在使用虚拟化环境时，云系统会面临如下风险：

(1) 如果主机受到破坏，那么主要的主机所管理的客户端服务器有可能面临被攻克的风险。

(2) 如果虚拟网络受到破坏，那么客户端也会受到损害。需要保障客户端共享和主机共享的安全，因为这些共享有可能被非法攻击者利用。

(3) 如果主机有问题，那么所有的虚拟机都会产生问题。

虚拟机和物理机之间的不同点将会逐渐减少。虚拟化带来的安全问题也只是才刚刚起步，虚拟环境中的安全机制与传统物理环境中的安全措施相比，仍有较大差距。所以，想要迁移至云计算环境中的用户需详细了解用户与云计算供应商所要承担的安全责任，安全的云计算环境是需要用户与供应商共同来维护的。

4) 云服务滥用

首先，云计算平台为用户提供低门槛的使用接口，用户不需要自行维护服务器、网络，只需专注于自身业务。用户可以通过互联网申请云计算平台中的资源，而且这种资源的使用是按需付费的，这就意味着使用资源进行信息发布与运营的能力是可扩展的，然而不法分子也同样可以利用云平台的这种能力，如利用庞大的网络资源和计算资源，组织大规模DDOS攻击，这样的攻击往往难以防范及溯源。可以预见的是，在云计算时代，随着系统规模的扩大和复杂性的提高，外部攻击也将更有效率，从而将传统的安全问题进一步放大，为安全防御带来更大的挑战。

其次，如果云计算服务被国外巨头垄断，这些垄断巨头及其背后的政治势力可以借此对我国进行远程监测和控制，并通过对用户整体情况进行统计分析，获取我国舆情动向、经济运行情况等重要数据。

同时，还可以有针对性地向我国推送反动有害等信息。这些都将对我国政治、经济、文化安全构成极大威胁。因此，应该将云平台视为重要的基础设施，应有专门机构对其运营情况进行监管：一方面要对云平台自身的滥用进行管理，另一方面要对云平台外部的攻击进行管理。

此外，由于对云服务可靠性和数据敏感性的担心，用户应优先选择本地的云供应商。鉴于目前国内云计算市场尚不成熟，政府和企业应继续制定相关技术标准，而这需要政府部门、研究机构、供应商、集成商和咨询公司等各方面的共同努力。

12.2　物联网系统及其安全问题

物联网的关键在于应用，物联网应用将深入到所有人生活的方方面面。物联网应用中所面临的安全威胁以及安全事故所造成的后果，将比互联网时代严重得多。物联网安全呈现大众化、平民化特征，安全事故的危害和影响巨大；物联网应用中各处都需要安全，安全措施与成本的矛盾十分突出。物联网安全，还必须改变先系统后安全的思路，在物联网应用设计和实施之初，就必须同时考虑应用和安全，将两者从一开始就紧密结合，系统地考虑感知、网络和应用的安全，才能更好地解决各种物联网安全问题，应对物联网安全的新挑战。

1. 物联网体系结构

物联网(Internet of Things，IoT)概念是 1999 年提出的，目前还没有权威的物联网定义。目前认可度比较高的物联网定义是：利用无线射频识别(Radio Frequency Identification Devices，RFID)、二维码、传感器、激光扫描器等各种感知技术和设备，将网络和所有物体相连，全面获取真实世界的各种信息，完成人与物、物与物的信息交互，以实现对现实世界物体的智能化识别、跟踪定位和管理控制。

从技术上讲，物联网是互联网的延伸和发展，不是全新凭空而出的。物联网是一个基于感知技术，融合了各类应用的服务型网络系统，可以利用现有各类网，通过自组网能力，无缝连接融合形成物联网。物联网体系结构包含 3 个层次，如图 12.2 所示，下层是感知真实世界的感知层，中间是完成数据传输的网络层，上层是面向用户的应用层。

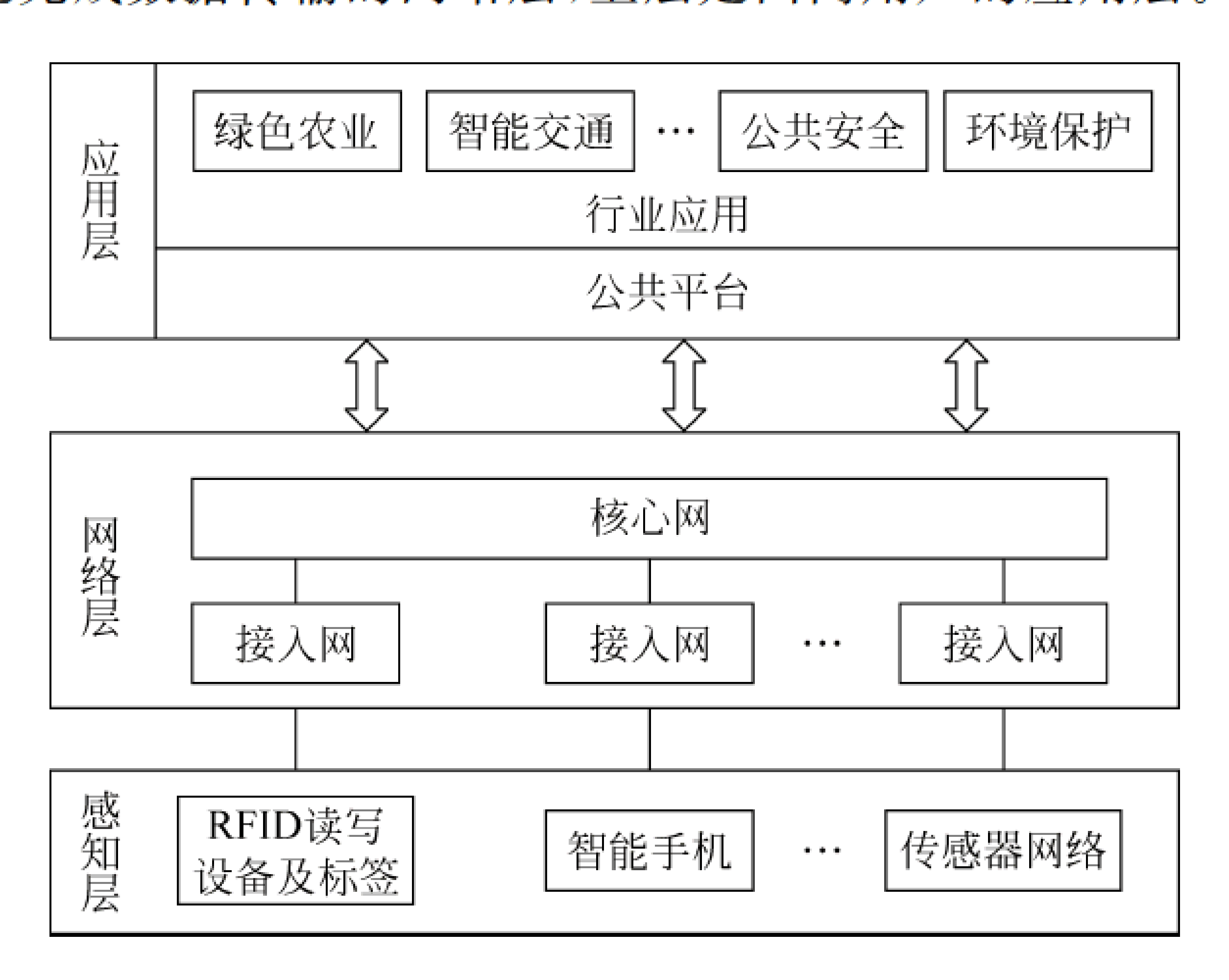


图 12.2　物联网体系结构图

2. 物联网安全新特征

与传统网络相比，物联网发展带来的安全问题将更为突出，要强化安全意识，把安全放在首位，超前研究物联网产业发展可能带来的安全问题。物联网安全除了要解决传统信息安全问题之外，还需要克服成本、复杂性等新的挑战。物联网安全面临的新挑战主要包括需求与成本的矛盾，安全复杂性进一步加大，信息技术发展本身带来的问题，以及物联网系统

攻击的复杂性和动态性仍较难把握等方面。总的来说,物联网安全的主要特点呈现 4 个方面,即大众化、轻量级、非对称和复杂性。

1）大众化

物联网时代,当每个人习惯于使用网络处理生活中的所有事情的时候,当人们习惯于网上购物、网上办公的时候,信息安全就与人们的日常生活紧密地结合在一起了,不再是可有可无的。物联网时代如果出现了安全问题,那每个人都将面临重大损失。只有当安全与人们的利益相关的时候,所有人才会重视安全,也就是所谓的“大众化”。

2）轻量级

物联网中需要解决的安全威胁数量庞大,并且与人们的生活密切相关。物联网安全必须是轻量级、低成本的安全解决方案。只有这种轻量级的思路,普通大众才可能接受。轻量级解决方案正是物联网安全的一大难点,安全措施的效果必须要好,同时要低成本,这样的需求可能会催生出一系列的安全新技术。

3）非对称

物联网中,各个网络边缘的感知节点的能力较弱,但是其数量庞大,而网络中心的信息处理系统的计算处理能力非常强,整个网络呈现出非对称的特点。物联网安全在面向这种非对称网络的时候,需要将能力弱的感知节点安全处理能力与网络中心强的处理能力结合起来,采用高效的安全管理措施,使其形成综合能力,从而能够整体上发挥出安全设备的效能。

4）复杂性

物联网安全十分复杂,从目前可认知的观点出发可以知道,物联网安全所面临的威胁、要解决的安全问题、所采用的安全技术,不仅在数量上比互联网大很多,而且还可能出现互联网安全所没有的新问题和新技术。物联网安全涉及到信息感知、信息传输和信息处理等多个方面,并且更加强调用户隐私。物联网安全各个层面的安全技术都需要综合考虑,系统的复杂性将是一大挑战,同时也将呈现大量的商机。

3. 物联网安全威胁分析

物联网各个层次都面临安全威胁,现分别从感知层、网络层和应用层对其面临的安全威胁进行分析。

1）感知层安全威胁

如果感知结点所感知的信息不采取安全防护或者安全防护的强度不够,则很可能这些信息被第三方非法获取,这种信息泄密某些时候可能造成很大的危害。由于安全防护措施的成本因素或者使用便利性等因素,很可能某些感知结点不会或者采取很简单的信息安全防护措施,这样将导致大量的信息被公开传输,其结果很可能是在意想不到的时候引起严重后果。

感知层普遍的安全威胁是某些普通结点被攻击者控制之后,其与关键结点交互的所有信息都将被攻击者获取。攻击者的目的除了窃听信息外,还可能通过其控制的感知结点发出错误信息,从而影响系统的正常运行。感知层安全措施必须能够判断和阻断恶意结点,并且还需要在阻断恶意结点后,保障感知层的连通性。

2）网络层安全威胁

物联网网络层的网络环境与目前的互联网网络环境一样,也存在安全挑战,并且由于其

中涉及到大量异构网络的互联互通，跨网络安全域的安全认证等方面会更加严重。

网络层很可能面临非授权结点非法接入的问题，如果网络层不采取网络接入控制措施，就很可能被非法接入，其结果可能是网络层负担加重或者传输错误信息。

互联网或者下一代网络将是物联网网络层的核心载体，互联网遇到的各种攻击仍然存在，甚至更多，需要有更好的安全防护措施和抗毁容灾机制。物联网终端设备处理能力和网络能力差异巨大，应对网络攻击的防护能力也有很大差别，传统互联网安全方案难以满足需求，并且也很难采用通用的安全方案解决所有问题，必须针对具体需求而制定多种安全方案。

3）应用层安全威胁

物联网应用层涉及到方方面面的应用，智能化是重要特征。智能化应用能够很好地处理海量数据，满足使用需求，但如果智能化应用一旦攻击者被利用，则将造成更加严重的后果。应用层的安全问题是综合性的，需要结合具体的应用展开应对。

4. 物联网安全技术体系结构

物联网安全需要对物联网的各个层次进行有效的安全保障，以应对感知层、网络层和应用层所面临的安全威胁，并且还要能够对各个层次的安全防护手段进行统一的管理和控制。物联网安全体系结构如图 12.3 所示。

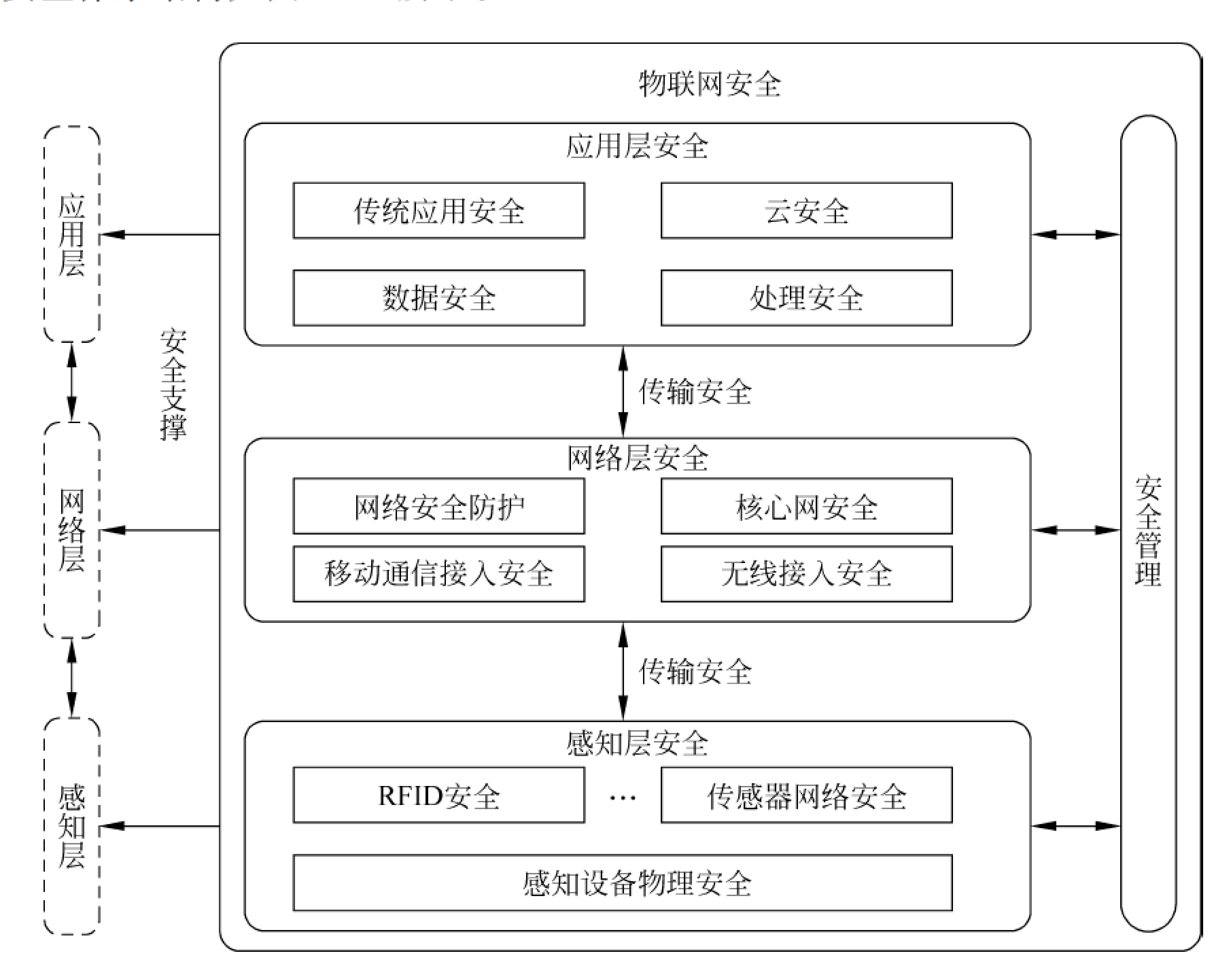


图 12.3 物联网安全体系结构

感知层安全主要分为设备物理安全和信息安全两类。传感器节点之间的信息需要保护，传感器网络需要安全通信机制，确保结点之间传输的信息不被未授权的第三方获得。安全通信机制需要使用密码技术。传感器网络中通信加密的难点在于轻量级的对称密码体制和轻量级加密算法。感知层主要通过各种安全服务和各类安全模块，实现各种安全机制，对

某个具体的传感器网络,可以选择不同的安全机制来满足其安全需求。

网络层安全主要包括网络安全防护、核心网安全、移动通信接入安全和无线接入安全等。网络层安全要实现端到端加密和节点间的数据加密。对于端到端加密,需要采用端到端认证、端到端密钥协商、密钥分发技术,并且要选用合适的加密算法,还需要进行数据完整性保护。对于结点间的数据加密,需要完成结点间的认证和密钥协商,加密算法和数据完整性保护则可以根据实际需求选取或省略。

应用层安全除了传统的应用安全之外,还需要加强处理安全、数据安全和云安全。多样化的物联网应用面临各种各样的安全问题,除了传统的信息安全问题,云计算安全问题也是物联网应用层所需要面对的。因此应用层需要一个强大而统一的安全管理平台,否则每个应用系统都建立自身的应用安全平台,将会影响安全互操作性,导致新一轮安全问题的产生。除了传统的访问控制、授权管理等安全防护手段,物联网应用层还需要新的安全机制,例如对个人隐私保护的安全需求等。

12.3 移动互联网安全技术

伴随着移动通信和互联网的发展,两者结合的产物移动互联网越来越受到人们的重视。通信产业巨头们纷纷确立自己的移动互联网发展战略,并推出了相关的产品。伴随着移动互联网的发展,有一些问题诸如安全问题也亟待解决。本节首先从移动互联网的发展引入,阐述移动互联网的特点以及它存在的安全威胁,再从安全隐患谈到相应的安全机制,最后就产业链协同发展、共同保护移动互联网的安全问题提出相关的建议。

1. 移动互联网的产生和发展及安全性方面特点

移动互联网(Mobile Internet)成为当今网络的一大热门词汇,其日趋火热也是网络发展的必然结果。目前,移动代替固定成为一个不可争辩的事实,而互联网的飞速发展,也是信息社会发展的一个必然趋势,两者的结合体移动互联网的诞生和发展就成为一个必然的产物。

从2006年下半年到现在,国际、国内一些ICT巨头的发展战略和重大事项,越来越注重于移动互联网。2006年9月,爱立信、诺基亚、GSM协会、微软、Google、沃达丰等13家公司共同创建mobi,宣告向移动互联网进军的决心;2007年,Intel在IDF大会上提出MID概念,而苹果则推出了代表性的终端iPhone,诺基亚则推出移动互联网服务品牌OVI,2008年Google推出了Android智能终端操作系统。ICT界的巨头纷纷推出自己的移动互联网战略,标志着一个崭新时代的开始。

移动互联网是以移动通信网作为接入网络的互联网及服务,它包括几个要素:移动通信网络接入,包括2G、3G和4G等;公众互联网服务;移动互联网终端。移动互联网和传统的互联网有一定的差异,从表现形式来看,传统互联网是开放的、免费的,拥有海量的信息平台,由于其本身设计理念,无法对用户进行身份确认,而移动互联网却能做到这一点。所以,相对而言,移动互联网对网络安全有着更高的要求,更强调保护用户的行为及隐私不受干扰。但是,目前移动互联网在终端和业务应用方面存在较大的安全漏洞,随着智能终端的普及,终端的安全和防护性存在重大考验;业务应用方面,使用移动互联网的用户可能会受到来自于各种渠道的垃圾信息,如短信、WAP、E-mail等,或者通过业务应用泄露用户信息。

相对而言,传统互联网上的 PC 终端防范意识起步较早,用户可以通过安装补丁或者杀毒软件进行终端和业务应用的保护。

2. 移动互联网安全威胁

移动互联网的安全通信框架可以参考 ITU-T X.805 框架,如图 12.4 所示,X.805 的安全框架基本上由 3 个层次、3 个平面和 8 个维度构成。从保障移动互联网的安全角度而言,我们认为要保证终端方面的安全、网络方面的安全和业务方面的安全,目前移动互联网在终端安全和业务安全方面相对存在较大安全隐患,需要人们注意。

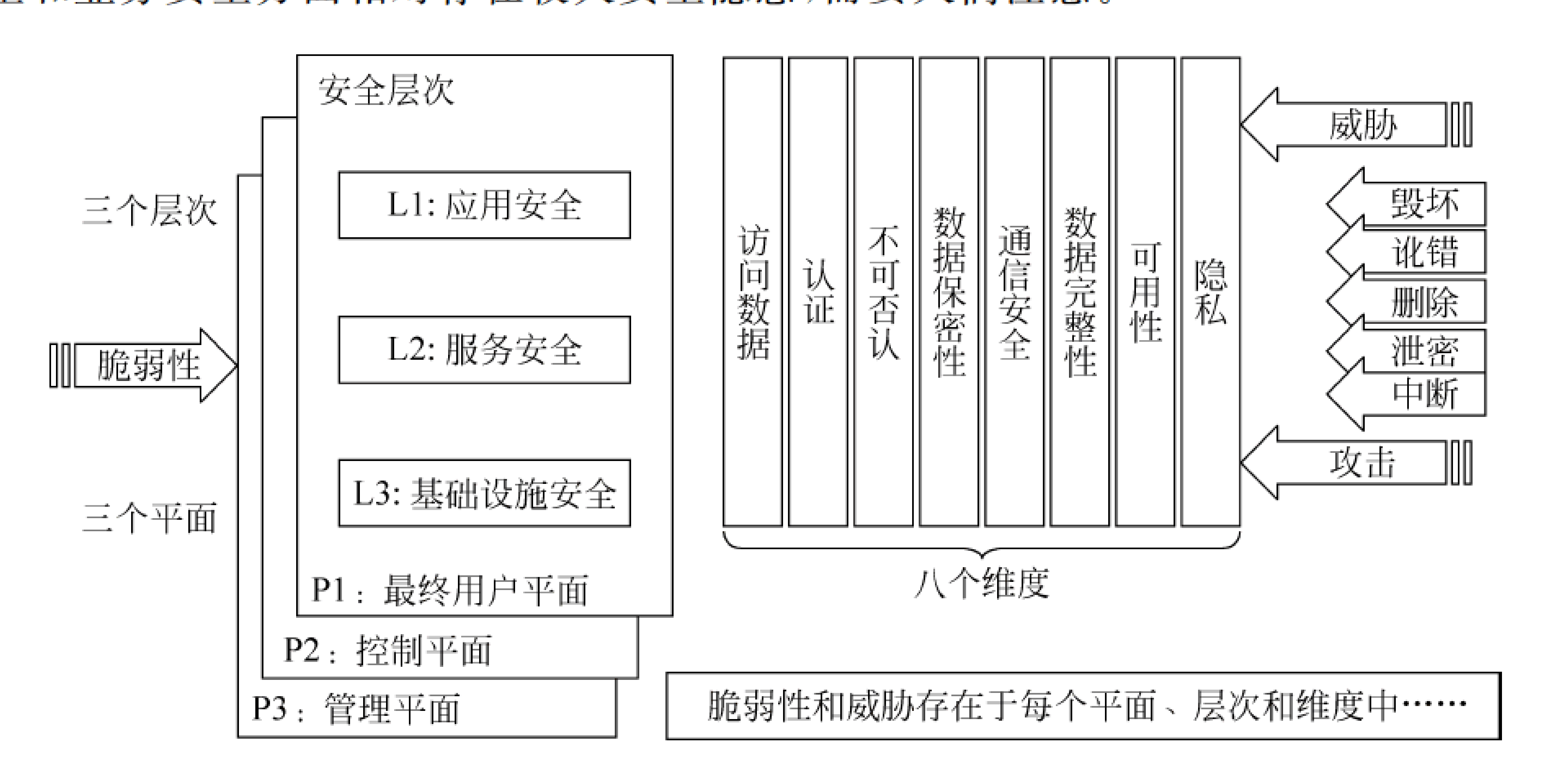


图 12.4 ITU-T X.805 安全通信框架图

1）终端方面的安全威胁

随着通信技术的进步,终端也越来越智能化,内存和芯片处理能力也逐渐增强,终端上也出现了操作系统并逐步开放。随着智能终端的出现,也带来了潜在的威胁,包括非法篡改信息、非法访问,或者通过操作系统修改终端中存在的信息、产生病毒和恶意代码进行破坏。

2）网络方面的安全威胁

在网络层面,存在进行非法接入网络,对数据进行机密性破坏、完整性破坏；进行拒绝服务攻击,利用各种手段产生数据包造成网络负荷过重等,还可以利用嗅探工具、系统漏洞、程序漏洞等各种方式进行攻击。

3）业务方面的安全威胁

在业务层面的安全威胁,包括非法访问业务、非法访问数据、拒绝服务攻击等。还包括目前存在的垃圾信息的泛滥、不良信息的传播,以及个人隐私和敏感信息的泄露,内容版权盗用和不合理的使用等问题。

3. 移动互联网安全机制

针对上述的安全威胁,移动互联网也有合适的安全机制来应对此类安全威胁。因为移动互联网接入部分是移动通信网络,无论是采用 2G、3G 或 4G 进行接入,3GPP、OMA 等组织都制定了完善的安全机制。我们也可以从终端、网络和业务的安全机制方面来分别进行阐述,通过相应的安全机制,能较好地控制移动互联网相关的不安全因素。

1）终端方面的安全机制

移动互联网终端，应具有身份认证的功能，具有对各种系统资源、业务应用的访问控制能力，对于身份认证可以通过口令方式或者智能卡方式、实体鉴别机制等手段保证安全性。对于数据信息的安全性保护和访问控制，可以通过设置访问控制策略来保证其安全性，对于终端内部存储的一些数据，可以通过分级存储和隔离，以及数据的完整性的检测等手段来保证安全性。

2）网络方面的安全机制

目前，在移动互联网接入网方面，无论是2G还是3G、4G，都有一套完整的安全机制。2G主要有基于时分多址（TDMA）的GSM系统、DAMPS系统及基于码分多址（CDMA）的CDMA 1X/2000系统，这两类系统安全机制的实现有很大区别，但都是基于私钥密码体制，采用共享秘密数据（私钥）的安全协议，实现对接入用户的认证和数据信息的保密，在身份认证及加密算法等方面存在着许多安全隐患；3G、4G在2G的基础上进行了改进，继承了2G系统安全的优点，同时针对3G、4G系统的新特性，定义了更加完善的安全特征与安全服务。目前3G、4G移动通信网络的安全机制包括3GPP和3GPP2两个类别。

3）业务方面的安全机制

对于业务方面，3GPP和3GPP2都有相应业务标准的机制。例如WAP安全机制、Presence业务安全机制、定位业务安全机制、移动支付业务安全机制等；其他方面还包括垃圾短消息的过滤机制，对于版权有OMA的DRM的标准等。移动互联网业务纷繁复杂，需要通过多种手段，不断健全业务方面的安全机制。

12.4 小结

云计算、物联网和移动互联网等新技术快速发展，基于这些新技术构建的信息系统也在各行各业得到越来越广泛的运用。传统的信息系统安全技术已不能完全适用于这些新型信息系统的安全防护工作，学习和研究针对这些新型信息系统的安全防护技术迫在眉睫。

本章对信息系统安全新技术进行总体介绍，详细介绍云计算系统的基本概念、面临的主要安全威胁和主要应对策略；讲述了物联网的概念、体系结构和安全新特征；介绍移动互联网的发展现状、安全性方面的特点以及主要安全机制。

通过本章内容的学习，读者可以了解信息系统的新技术、新技术面临的安全威胁，主要的安全防护技术等内容。

习题

一、填空题

1. 云计算安全包括两个主要研究方向，分别是________和________。

2. ________是云计算的重要特色，________有效加强了基础设施、平台、软件层面的扩展能力。

3. 物联网体系结构包含 3 个层次,分别是________、________和________。

4. 移动互联网面临的安全威胁主要来自 3 个方面,分别是________、________和________。

二、选择题

1. 云计算的三种交付模式是 IaaS、PaaS 和________。

A. BaaS　B. SaaS　C. TaaS　D. HaaS

2. ________是一个基于感知技术,融合了各类应用的服务型网络系统,可以利用现有各类网,通过自组网能力,无缝连接融合形成。

A. 互联网　B. 移动互联网　C. 物联网　D. 无线自组网

三、简答题

1. 试述云计算面临的主要安全问题及应对策略。

2. 物联网安全的新特征有哪些?

3. 移动互联网的安全机制有哪些?

参 考 文 献

[1] 熊平,朱天清.信息安全原理及应用.北京:清华大学出版社,2009.
[2] 陈波.计算机系统安全原理与技术(第3版).北京:机械工业出版社,2013.
[3] 俞承杭.信息安全技术(第二版).北京:科学出版社,2011.
[4] 冯登国,赵险峰.信息安全技术概论.北京:电子工业出版社,2014.
[5] 沈昌祥.信息安全导论.北京:电子工业出版社,2009.
[6] 卿斯汉,沈晴霓,刘文清.操作系统安全(第2版).北京:清华大学出版社,2011.
[7] 吴世忠,江常青,彭勇.信息安全保障基础.北京:航空工业出版社,2009.
[8] 李剑,张然.信息安全概论.北京:机械工业出版社,2014.
[9] 曹天杰,张永平,毕方明.计算机系统安全.北京:高等教育出版社,2007.
[10] 范红,冯登国.信息安全风险评估实施教程.北京:清华大学出版社,2007.
[11] 吴晓平,付珏.信息安全风险评估教程.武汉:武汉大学出版社,2011.
[12] 吴亚非,李新友,禄凯.信息安全风险评估.北京:清华大学出版社,2006.
[13] 郭亚军,宋建华,李莉等.信息安全原理与技术.北京:清华大学出版社,2008.
[14] 马建峰 沈玉龙.信息安全.西安:西安电子科技大学出版社,2013.
[15] 孙钟秀,费翔林.操作系统教程.北京:高等教育出版社,2008.
[16] 宋金玉,陈萍.数据库原理与应用.北京:清华大学出版社,2011.
[17] Willam Stalling.密码编码学与网络安全:原理与实践.北京:电子工业出版社,2006.
[18] 陈越,寇红召,费晓飞等.数据库安全.北京:国防工业出版社,2011.
[19] 王斌君,景乾元,吉增瑞等.信息安全体系.北京:高等教育出版社,2008.
[20] 张娜.分布式网络安全审计系统.上海:华东师范大学硕士学位论文,2009.
[21] 黄志国.数据库安全审计的研究.太原:中北大学硕士学位论文,2006.
[22] 赖丽.基于Oracle的数据库安全审计技术研究.成都:四川师范大学硕士学位论文,2009.
[23] 逯楠楠.数据库安全审计分析技术研究与应用.武汉:湖北工业大学硕士学位论文,2011.
[24] 吴纪芸,陈志德.数据库安全评估方法研究.中国科技信息,2015(02):108~110.
[25] 张敏.数据库安全研究现状与展望.中国科学院院刊,2011.3(26):303-309.
[26] 成明盛.数据库备份恢复技术的研究及应用设计.成都:西南交通大学硕士学位论文,2002.
[27] 裴小燕,张尼.浅谈云计算安全,信息通信技术,2012(01):24~28.
[28] 孙建华,陈昌祥.物联网安全初探.通信技术,2012.07(45):100~102.
[29] 马军,马慧.移动互联网安全问题分析与建议,现代电信科技,2009.7(7):46~49.